ARMY RESEARCH LABORATORY

Default Operational Representations of Military Organizations

Samuel C. Chamberlain

ARL-TR-2172 FEBRUARY 2000

Approved for public release; distribution is unlimited.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-2172 February 2000

Default Operational Representations of Military Organizations

Samuel C. Chamberlain
Information Science & Technology Directorate

Abstract

This report is a result of a study commissioned under the auspices of the C4ISR[a] Cooperative Research Program (CCRP), within the OASD(C3I)[b], to attempt to identify why information integration between battlefield systems has not progressed at the rate expected. It was observed that sometimes the most fundamental issues of battle command are overlooked, especially when automation is involved. This is because of one's natural inclination to automate manual procedures rather than exploit the true capabilities of new technologies. This report addresses one of these cases: the identification of many entities inside many interconnected computers, sometimes called the "naming problem." It is argued that this problem strikes at the heart of battle command automation process and, consequently, the development and execution of mission capability packages. However, one must not be deluded into believing that this is merely a computer science problem; it is primarily a military science problem with some computer science technology "sprinkled in." The thesis is presented that the concept of organization (or task organization) is the central around which all battle command representations revolve. In essence, the organizational structure forms a framework to which all other battlefield entities can be related, making the organizational data structures the rallying point for the integration of other databases, such as logistics, personnel, and communications. Further, it is shown that fluid orders of battle (OOBs) can most always be built by re-linking existing organizations that are part of a stable default organizational structure. For this to be effective, however, the default structure must include more nodes than provided by current staffing documents. The concept of *default operational organizations* (DOOs) is introduced to provide a representation of military organizations that meets the requirements necessary to build arbitrary OOBs across joint service and international boundaries. By closely studying how each service organizes for combat, the author developed several basic tenets in an attempt to reduce many practices to a few fundamental concepts. The result is a set of guidelines based upon the "best practices" of all the services, which is used to build DOOs that closely reflect the way we fight or deploy. An architecture is described to maintain the DOO in organization servers, and a unique identifier is proposed, called an organization identifier (Org-ID), to facilitate the naming process. Org-IDs are distributed by a set of Org-ID servers that guarantee the uniqueness of these identifiers across service and coalition boundaries. Finally, a naming convention is introduced that builds upon the uniqueness of Org-IDs to provide universal surrogate keys for any battlefield entity.

---

[a] C4ISR: command, control, communications, computers, intelligence, surveillance, and reconnaissance

[b] OASD(C3I): Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence

ACKNOWLEDGMENTS

This study was sponsored and funded in part by the Department of Defense (DoD) C4ISR[a] Cooperative Research Program (CCRP), (http://www.dodccrp.org) which is managed by Dr. David Alberts (david.alberts@osd.pentagon.mil), Director of Research for the Office of the Assistant Secretary of Defense for Command Control, Communications, and Intelligence (OASD[C3I]).

The following organizations and people contributed valuable operational advice during this study:

- U.S. Army Training and Doctrine Command (TRADOC) Program Integration Office (TPIO) for Army Battle Command Systems (ABCS), Fort Leavenworth, KS. In particular,

    COL James Bessler (Director 1998-1999),
    COL John Bartley (Director)
    LTC Robert Hartel,
    MAJ Mike Boller, and
    MAJ Robert Savage (USMC).

- The US Air Force Aerospace C4ISR Center, Langley Air Force Base, VA. In particular,

    Mr. Garry Barringer, (Technical Director),
    MAJ Bryan Bearden, and
    LTC Robert Jordan.

- The U.S. Navy N3, the Pentagon. In particular,

    LCDR Patrick Dunn.

- The U.S. Naval Warfare Development Center, Newport, RI. In particular,

    CDR Craig Geron, and
    LCDR Burt Espe.

- The U.S. Naval Postgraduate School, Monterey, CA. In particular,

    CDR Bill Hatch.

- The U.S. Marine Corps Combat Development Command, Quantico, VA. In particular,

    MAJ Vaughan Fox.

[a] C4ISR: command, control, communications, computers, intelligence, surveillance, and reconnaissance.

INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Page

LIST OF FIGURES ... vii
LIST OF TABLES ... ix
1 INTRODUCTION ... 1
1.1 What is an Organization? ... 2
1.2 A Simple Data Abstraction for Organizational Structure ... 3
1.2.1 Data Storage, Identifiers and Universal Surrogate Keys ... 6
1.2.2 Organization-to-Organization Relationships (OTOR) and Roles ... 8
1.2.3 Allocation of Organization Identifiers (Org-IDs) ... 10
1.3 The Organization Server Architecture ... 10
1.3.1 Organization Servers Versus Org-ID Servers ... 11
1.3.2 Org-ID Stability ... 13
2 DEFAULT OPERATIONAL ORGANIZATIONS ... 15
2.1 Billets and Leaders ... 16
2.1.1 Leadership Billets ... 17
2.1.2 Chain of Command Versus Command Structure ... 19
2.2 Crews ... 22
2.2.1 Assignments, OTORs, and Organization Trees ... 25
2.2.2 Alignments, ETORs, and Organization Trees ... 27
2.3 Doctrinal Organization ... 31
2.3.1 Importance of Doctrinal Organizations ... 35
2.4 Putting the Pieces Together ... 36
2.4.1 Default Operational Organization Examples ... 36
2.4.1.1 Aviation Example – A Strike Package ... 36
2.4.1.2 Naval Examples – Administrative and Operational Command Structures ... 41
2.4.1.3 Army Examples – Aviation Packages ... 45
3 UNIVERSAL NAMING ... 48
3.1 Special Cases: Enemy and Ad Hoc Organizations ... 52
3.1.1 Enemy Organizations ... 52
3.1.2 Ad Hoc Organizations ... 52
4 CONCLUSION ... 55
5 RECOMMENDATIONS ... 58
REFERENCES ... 59
APPENDICES
A. Current DoD and NATO Definitions Related to Organizational Relationships ... 61
B. Approximately 80% of Nodes Are Leaves ... 69
C. Depth-First Search Algorithm to Derive a Chain of Command from a Command Structure ... 73
DISTRIBUTION LIST ... 79
REPORT DOCUMENTATION PAGE ... 83

INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure		Page
1	Graph Definitions and Terms	3
2	Two Chains of Command Displayed Simultaneously	5
3	An Irregular Organization Chart	5
4	Computer Representation of an Organization Chart	7
5	Temporary Nodes Versus Permanent Links	9
6	Organization Server Architecture	11
7	Creating a New Organization	12
8	Identifying Unknown Org-IDs	12
9	Task Organizing Does Not Require the Creation of New Organizations	14
10	Billets Ultimately Form the Leaves of the Trees	17
11	Nested Leadership Billets	18
12	Abstract View of Nested Leadership Billets	18
13	A Command Structure Depicts Decomposition	19
14	Chain of Command Derived From Command Structure	20
15	Non-nested Command Structure	21
16	Three Categories of Relationships: OTORs, PTORs, and ETORs	23
17	Example of Habitual and Non-Habitual Relationships	24
18	Four Combinations of Crew Relationships	25
19	Habitual Assignments and Alignments	26
20	Non-Habitual Relationships	26
21	System-Oriented Approach to Non-Habitual Alignments	28
22	System Organization With Predefined Roles (links)	29
23	Crew-Oriented Approach to Non-Habitual Alignments	30
24	Two Extreme Cases of a Tree With 16 Leaf Nodes	31
25	Doctrinal Organizations Missing From Staffing Documents	33
26	Example of Doctrinal Organizations	34
27	Building a Simple Strike Package	37
28	System-Based DOO for a Strike Package	39
29	Strike Package 53 Split During Operations	40
30	Membership in Multiple Command Structures	41
31	A Simplified View of a Task Organized Amphibious Force	42
32	An Asset With Multiple Habitual Crews	43
33	Administrative Structure from an SMD	44
34	Operational Command Structure - Battle Bill (Watch) Control Stations	45
35	Current Army TO&E Structure of an Aviation Company	46
36	A Notional DOO of an Army Aviation Company	47
37	Using a DOO to Build an Order of Battle	48
38	Surrogate Keys Composed of Composite Integers	49
39	Ultimately, Default Data Available for Download from Many Sources	50
40	Universal Surrogate Keys Are Interchangeable	55

INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table **Page**

1. Relational Database Table Terms 6
2. Common DoD and NATO OTOR Types 8
3. Database Tables Showing Named Links, or "Roles" 9

INTENTIONALLY LEFT BLANK

1 INTRODUCTION

Military databases are composed of numerous, different entities that collectively represent a model, or data abstraction, of the battlefield or other subjects of interest. These entities include representations of both physical items, such as equipment and personnel, and conceptual items, such as plans and organizations. This report discusses the importance of the concept of an "organization" and its relationship to other entities. The central theme is that the formal concept of organization forms the framework to which all other battle command entities and functions relate, and this can be exploited to solve several problems. In essence, the organizational structure that defines a military unit forms a framework to which all other battlefield entities are ultimately attached, making the organizational data structures the rallying point for the integration of other databases, such as logistics, personnel, communications, and planning.

Currently, there are numerous approaches for identifying organizations and this is hindering a primary objective: the capability to "Plug and Play" task organizations across international and service boundaries, especially while building military capability packages (MCPs); for a discussion of MCPs, see Alberts (1995). To achieve ultimate interoperability, every entity, not just organizations, must be universally named. Therefore, any identification strategy must be globally unique and consistent. This report describes how the concept of organization can be used as a foundation to address the complete "naming problem" between military and civilian distributed systems and data entities. However, one must not be deluded into believing that this is merely a computer science problem; it is primarily a military science problem with some computer science technology "sprinkled in." Thus, a major portion of this report concerns issues regarding the formal representation of military and related organizations.

Two hypotheses are introduced:

Hypothesis 1: "Default" (or administrative) command structures exist that are relatively stable and, if designed carefully, can be used as the base structure for integrating entities and building orders of battle (OOBs).

Hypothesis 2: Operational command structures are fluid and are nearly always constructed by modifying (i.e., re-linking) the default command structures.

Corollary 2a: When building an operational command structure (for an OOB), one should rarely have to create new organizations. However, one may want to provide an alias for an existing organization.

Therefore, OOBs can be created and disseminated by exchanging lists of command structure links, i.e., via pairs of numeric values that identify an organization and its new parents. This can be very dynamic, even though the existence of the base organizations is typically stable.

1.1 What is an Organization?

There are several definitions for "organization" or "unit[1]," for example,

Webster's: *organization*—an administrative and functional structure[2], or

Joint Services Dictionary: *unit* (Department of Defense [DoD], North Atlantic Treaty Organization [NATO]) — 1. Any military element whose structure is prescribed by competent authority, ... specifically, part of an organization[3],

However, a rigorous, formal definition is required to address the myriad issues involved when one attempts to automate a battle command process (such as MCP development). Over the past several years, formal data modeling tools and processes have permeated the database development community. One notable example is the relational IDEF1X[4] information modeling method that introduces "business rules" as an approach that has provided a tremendous improvement in the data modeling process (see Bruce 1992). Both Booch and Rumbaugh have defined analogous approaches for the object-oriented environment (see Booch 1994; Rumbaugh et al. 1991). However, even with the introduction of more rigorous modeling tools, many procedural issues that need to be addressed are still undefined. The discipline of logic programming may someday help to define some of this information (see Minker 1996; Grant & Minker 1992; Robinson 1992), but formal military science is required to first describe what it is one does during the "task organizing" process.

It is important to understand that an organization is a virtual entity; that is, one cannot touch an organization. In its simplest state, it is mental clustering of real-world objects, collected together, based upon human thoughts. In the military, an organization typically manifests itself as an "organization chart," computer listings, or database entries that describe people and equipment "assigned" to the organization. In this sense, the concept of an organization forms the basis of command. Before one can command, there must be something to command; thus, organizations are formed, either formally or informally, to serve as aggregation points for functional and command purposes. The rules for building these structures are often elusive, however, when the formal definition process begins (as the author has discovered from discussions with military experts). It is easy to build structures which, although meaningful to a human, are not clear when represented inside a computer that is expected to automatically redirect operations and reorganize processes in the midst of dynamic (and sometimes catastrophic) events.

"Organigraphs," introduced by Mintzberg and Van der Heyden (1999), provide a fresh way to document organizational structure, which broadens the definition to include interactions, processes, products, and information exchange requirements. The topological structures of sets, chains, hubs, and webs are used to describe the many facets encountered as organizations operate in their daily environments. This report focuses on one microcosm of the topic of organizations, the representation of the military command structure. The formalism presented is completely general and may be used to describe any of the organigraph structures. However, because of the properties of military command and its focus on the commander, this report centers on the attributes of the "tree structure" (a special type of chain) as it is used to represent the military command structure.

---

[1] Informally, the terms organization and unit have been synonymous.

[2] *Webster's 10th Collegiate Dictionary*

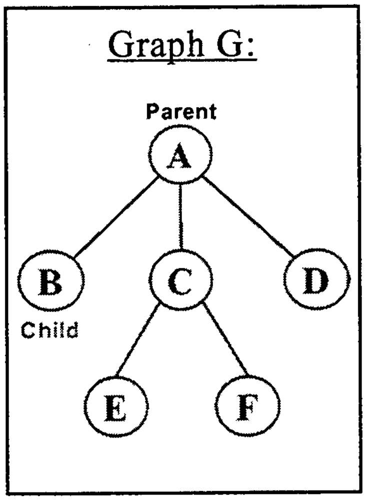
[3] http://www.dtic.mil/doctrine/jel/doddict

[4] IDEF1X: ICAM (integrated computer aided manufacturing) **DEF**inition method number **1** – e**X**tended.

1.2 A Simple Data Abstraction for Organizational Structure

Although informal interactions are common, military organizations ultimately manifest themselves as organization charts that describe the aggregation and composition of clusters of people and equipment. To formalize the process of building organization charts, we present them in terms of graph theory. A graph is composed of a set of ***nodes*** connected by a set of ***links*** (i.e., vertices {V} and edges {E}, respectively). A ***tree*** is a special graph that is *fully connected* (i.e., every node is linked to at least one other node) and there are *no cycles* (i.e., when links are traversed, only one path exists between any two nodes). Normally, one node is selected as the "beginning" of the tree and is named the root node. Figure 1 summarizes several graph theory terms.

By *definition*, nodes represent ***organizations*** (e.g., A, B, C, D, E, and F in Figure 1), and links represent command relationships (e.g., { (A,B), (A,C), (A,D), (C,E), (C,F) }), the set of which defines a ***command structure***.[5] Now, a new (and perhaps controversial) definition for ***unit*** is introduced: *a unit is **graph** that is composed of both a set of organizations (nodes) and a command structure (set of links)*. Therefore, in Figure 1, Graph G equates to Unit G, and it is composed of six organizations and a command structure of five command relationships. Previously, the term organization was ambiguous and it could be considered a node or a whole graph composed of a root node and its descendants. Now, these terms are formally defined with very specific, albeit slightly different, formal meanings or *semantics* (the meaning of the data or the concepts behind the data).



***NODES* (or *vertices*): set V = { A, B, C, D, E, F }**

***LINKS* (*edges*): set E = { (A,B), (A,C), (A,D), (C,E), C,F) }**

***GRAPH*: collection of vertices and edges: G(V,E)**

A *Tree* structure is a "connected" graph with no "cycles," i.e., every node has at least one link to another node and only one path exists between any two nodes.

Via a link, a node can be a *parent* or a *child* of another node.

A node without a child is called a *terminal or leaf node* (e.g., the nodes at the bottom of the tree: B, D, E, and F)

A node with children is an non-terminal or *internal node* (e.g., A and C);

The *root node* is a special internal node with no parent (e.g., A).

Organization Charts are Trees (w/ boxes instead of circles)
(Often the name of the tree is inherited from the name of the root node - e.g., A):

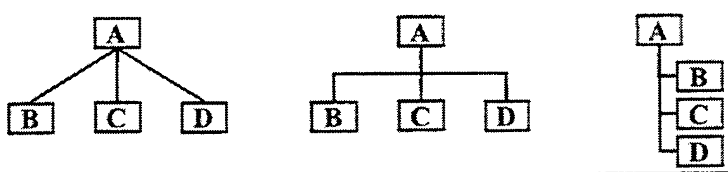


Figure 1: Graph Definitions and Terms

[5] Formally (mathematics), a relation is a set of ordered pairs (e.g., R = { (a,b), (b,c), (c,d) }), so this term is not used in these definitions. The term relationship is not a formal concept and will be used to describe an extended ordered pair.

The recommended semantics are

Organization: a single node of an organizational structure,
Command Relationship: a link (read "is composed of") between two nodes (organizations) with attributes to define the properties of the link; more formally

$$\mathbf{CR} = (\,(\mathbf{A},\mathbf{B}), \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\,)$$, in which (A,B) is a link between nodes A and B, and X_i's are attributes, for example, relationship type (T), effective times (Ti), role (R), etc.

Command Structure: a set of command relationships (links) that define the composition of a tree graph; more formally

$$\mathbf{CS} = \{\, \mathbf{CR}_1, \mathbf{CR}_2, \ldots, \mathbf{CR}_k \,\}$$, in which k is the number of links;

For example, in Figure 1,

$$\mathbf{CS}_G = \{\, \mathbf{CR}_1, \mathbf{CR}_2, \mathbf{CR}_3, \mathbf{CR}_4, \mathbf{CR}_5 \,\}$$
$$= \{\, (\,(\mathbf{A},\mathbf{B}), \mathbf{T}_1, \mathbf{Ti}_1, \mathbf{R}_1\,),\ (\,(\mathbf{A},\mathbf{C}), \mathbf{T}_2, \mathbf{Ti}_2, \mathbf{R}_2\,),\ (\,(\mathbf{A},\mathbf{D}), \mathbf{T}_3, \mathbf{Ti}_3, \mathbf{R}_3\,),\ (\,(\mathbf{C},\mathbf{E}), \mathbf{T}_4, \mathbf{Ti}_4, \mathbf{R}_4\,),\ (\,(\mathbf{C},\mathbf{F}), \mathbf{T}_5, \mathbf{Ti}_5, \mathbf{R}_5\,) \,\}$$

Unit: a graph composed of organizations (nodes) and command relationships (links),
Task Organization: the process of rearranging units (graphs).

Using these definitions, the terms unit and organization are *not* synonymous. Graph G represents a *unit* composed of organizations A through F connected via the command structure { (A,B), (A,C), (A,D), (C,E), (C,F) }. Confusion is introduced in the military domain because typically, a unit (a graph) inherits its name from the root organization (a node), such as Organization A in Figure 1. Although the difference between *Unit A* and *Organization A* may be ambiguous from a natural language perspective (e.g., English), it is not from a formal perspective. Organization A is a single node, whereas Unit A includes Organization A and all its descendants connected via a specific command structure that may descend to any arbitrary depth. Using these semantics, it is correct to ask the question, "Under the current command structure, what organizations are in Unit A?" (answer: A,B,C,D,E,F). *Therefore, modifying the command structure of a unit (i.e., the links to the nodes of a tree graph) modifies its structure or composition, but it does not change the individual organizations themselves.* Although this is a technical detail of these semantics, it is important to understand this distinction to prevent significant confusion as more complicated concepts are presented.

One of the goals of this project is to produce consistent and unambiguous organization charts, which means defining unambiguous command structures. In other words, there may be more than one command structure at any given time, but each command structure must be explicit and must produce a graph with tree properties. Figure 2 illustrates an organization chart that may appear confusing to those not familiar with the Navy practice of maintaining two distinct, simultaneous command structures: one for administrative purposes and one for operational purposes.[6] In Figure 2, both are displayed together, even though each is a separate explicit tree structure. Although this may appear ambiguous to some, it is not ambiguous to one experienced in the Navy or Marine Corps operations.

Figure 3, on the other hand, is an example of an organization chart with irregular features. The question arises whether this chart is constructed in this manner to emphasize that three different command relationship types are involved, or it is just the designer's style? For a computer to

[6] A formal definition for chain of command is introduced in Section 2.1.2 as part of the discussion of billets.

understand and display this chart, more information must be provided about the command relationships between the "children" nodes and their "parent" node (which in this diagram is also the root node).

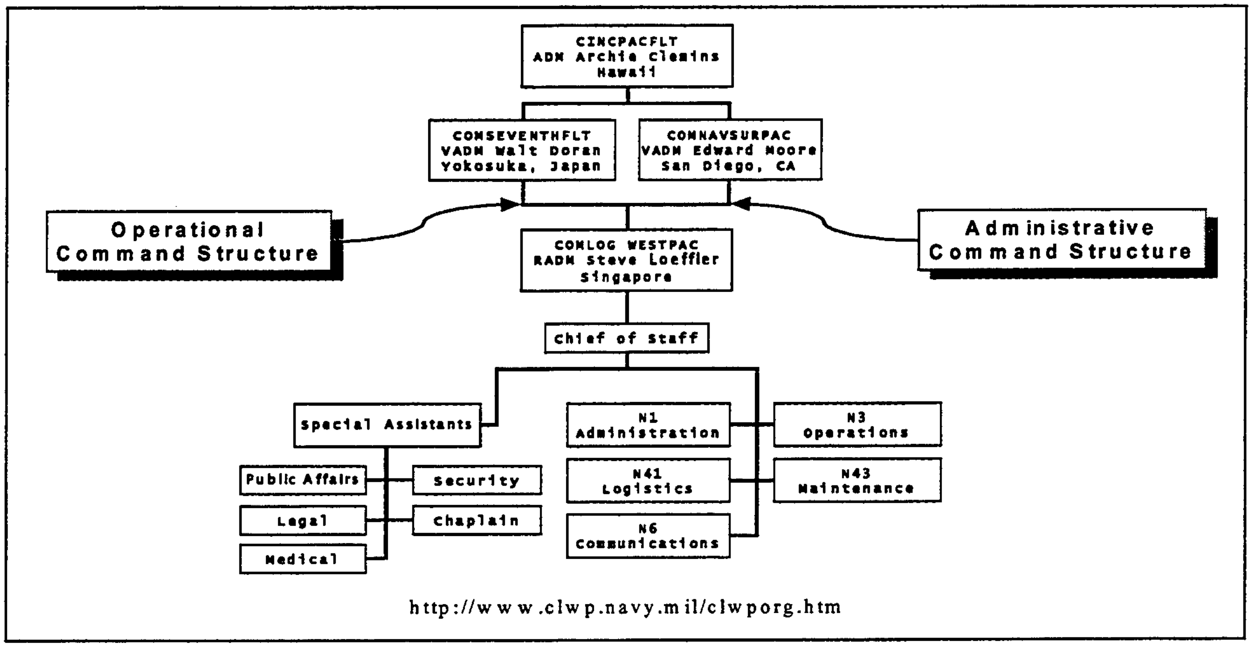


Figure 2: Two Chains of Command Displayed Simultaneously

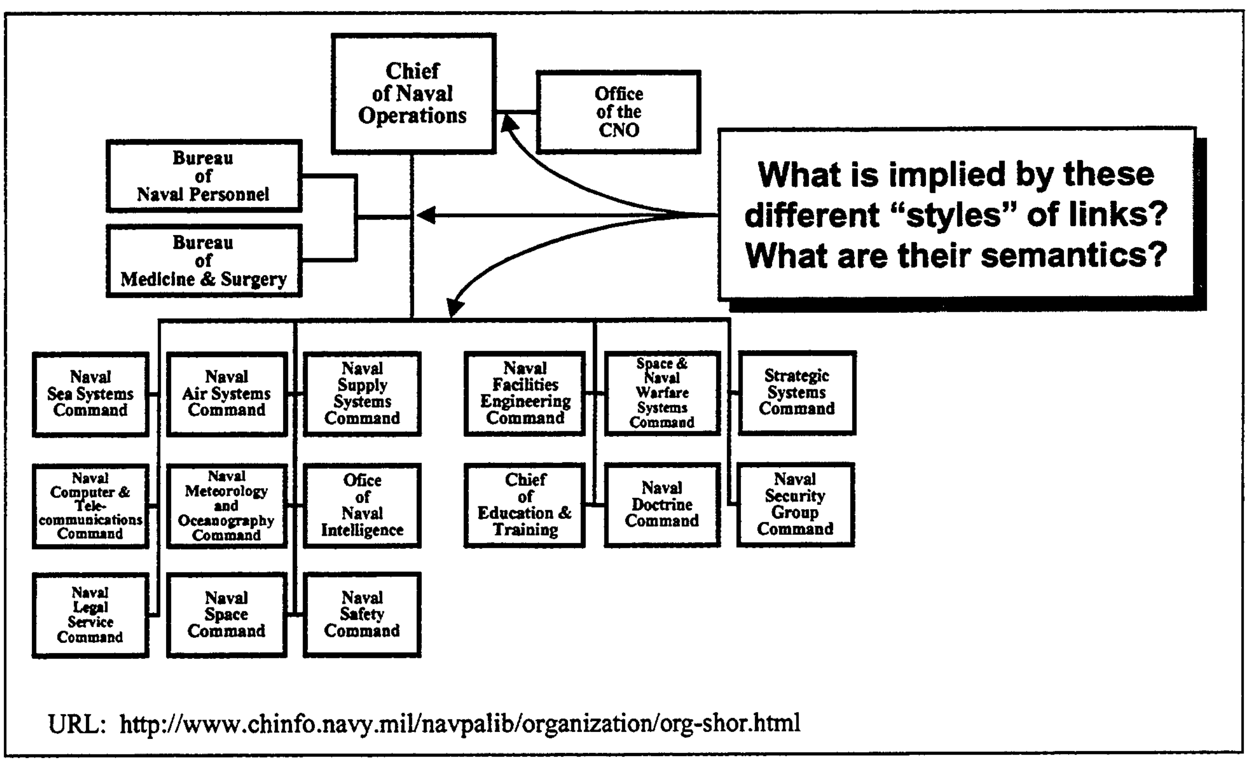


Figure 3: An Irregular Organization Chart

One must decide what constitutes being a bona fide organization and how to formally represent the links or relationships between them. This does not mean that every system has to display organization charts the same way but only that the organizational structure should be represented inside the computer, based on a common set of semantics. If this is done uniformly, one can begin to build applications that can "Plug and Play" task organization across service and even coalition boundaries. With this paradigm, the primary technique for accessing unit information is via tree traversal algorithms, such as depth-first and breadth-first search (Knuth 1973). With parent-child associations defined for every node, attributes such as "echelon" have little importance other than aiding display graphics.

1.2.1 Data Storage, Identifiers and Universal Surrogate Keys

Database technology provides a rigorous facility to store and maintain information about organizational structures.[7] A database is composed of *entities* that are described via *attributes*. Each single occurrence of an entity is called an *instance*. Physically, in a relational database, an entity corresponds to a table whose columns (or fields) correspond to the entity's attributes and whose rows are the individual instances. For example, in Table 1, the table or entity is named "Example" and has five attributes (A through E) that are the names of the table's columns (or fields). The table has eight entries or instances that are labeled Instance 1 through 8 (the fields are not complete). Every instance of a table must have a unique way of identifying it from every other instance. An *identifier* (also called a candidate key) is an attribute, or minimum set of attributes, that uniquely identifies a particular instance of an entity (i.e., row of a table). Since an entity may have several identifiers, a *primary key* is the principal identifier chosen to uniquely identify an entity. In practice, primary keys may be created by combining primary keys from other entities to ensure that they remain unique.[8] This practice is often illustrated with a video store example in which individual videotapes are identified with a primary key that is composed of a unique movie name, followed by a copy number (e.g., "A Bridge Too Far" – Copy 1).

Table 1: Relational Database Table Terms

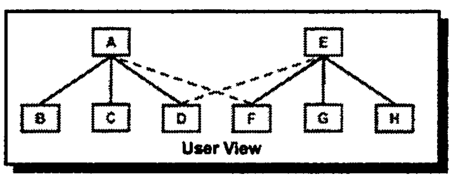
Table Example				
Attribute A	Attribute B	Attribute C	Attribute D	Attribute E
Instance 1	-	-	-	-
Instance 2	-	-	-	-
Instance 3	-	-	-	-
Instance 4	-	-	-	-
Instance 5	-	-	-	-
Instance 6	-	-	-	-
Instance 7	-	-	-	-
Instance 8	-	-	-	-

[7] Although this is true for other strongly typed, structured information standards (such as XML, the Extensible Markup Language), databases have the additional advantage that they are supported by extensive data modeling tools that allow the semantics of the information (sometimes called "business rules") to be included in the definition process.

[8] This process is referred to as using *identifying relationships.*

Although the term "naming convention" typically brings to mind human readable forms of creating and identifying entities, this is not necessary for computers. From a computing perspective, there is nothing simpler than an integer, and there are advantages to using integers for primary keys. A ***surrogate key*** is a primary key that has no inherent meaning, is composed of only a single (database) field, and whose value is not derived from any other entity (i.e., is non-identifying). Integers make good primary keys because they are simple, easy to manipulate by machines, and easy to verify for uniqueness.[9] For example, in Table 1, if the field named Attribute A contains surrogate keys, then the values of Instance 1 through Instance 8 would be integers that uniquely identify each of the eight rows of Table Example. Normally, users will never know that surrogate keys exist, but the database management system and application programs can use them extensively to greatly enhance performance and flexibility. Codd (1979) and Date (1986) provide early descriptions and the rationale for using surrogate keys, and Lonigro (1997) provides a simple description of the advantages of surrogate keys (see http://www.dbpd.com/vault/9805xtra.htm).

Entities are usually related to each other. For example, in Figure 1, a relationship *is_a_parent_of*, or more specifically, *is_composed_of*, exists between nodes A and B as well as between nodes A and C and nodes A and D (e.g., A *is_composed_of* B). Surrogate keys facilitate this process because they provide a simple, terse way to reference associations between entities; Figure 4 illustrates this process. At the top of the figure is a box containing a "user's view" of an organization chart similar to that in Figure 1. A second box depicts the corresponding database tables, one for nodes and one for links, used to represent the structure of the organization chart (a



Conceptually: Two Tables -
One for Organizations (or Nodes), and
One for the Command Structure (or Links).

Nodes

Organization Table	
Org-ID	Name
662	"A"
1253	"B"
23	"C"
775	"D"
22	"E"
8153	"F"
25	"G"
952	"H"

Links

Organization-Association Table					
Org-Assoc-ID	Child	Parent	Assoc Type	DTG	Role
25343	1253	662	Assign	-	Command
76524	23	662	Assign	-	-
22354	775	662	Assign	-	-
73642	8153	22	Assign	-	-
872	25	22	Assign	-	-
772433	952	22	Assign	-	-
943756	775	22	Attached	$T_1 : T_2$	Habitual
943757	8153	662	Attached	$T_1 : T_2$	Habitual

Computer View

Temporary Links Added to Local Databases Per Operations Plan.

Figure 4: Computer Representation of an Organization Chart.

[9] Technically, an identifier is just a fixed width bit pattern and not an integer. However, since it can be displayed as a whole number, the term integer is used for conceptual simplicity.

tree graph). The table labeled "Organization Table" represents the nodes of the graph and contains two fields: one for a surrogate key, called "Org-ID," to uniquely identify the table entry, and a second, called the "Name," that provides a familiar name for the node. A second table, called the "Organization-Association Table," contains the link information that describes how the different nodes of the graph (org chart) are related. In this table, there are five fields. The first is the "Org-Assoc-ID" field that uniquely identifies each table entry using a surrogate key. Each entry corresponds to one command relationship (link) and identifies, via surrogate keys, a parent and its child, and other attributes such as the type of relationship between them (e.g., the association type) and any constraints on the lifetime of the relationship (e.g., a date–time group [DTG] for the effective start time). In this example, links of type "Assign" indicate the default case that is permanent if not superseded by other temporary relationships.

It is perfectly admissible to re-link nodes on a temporary basis; this activity is called "task organizing" in the military domain. Consequently, the last two rows in the "Organization-Association Table" indicate that from time T_1 to time T_2, the nodes with the names "D" and "F" (identified by surrogate keys 775 and 8153) switch parents. After time T_2, the nodes revert to their default case. It is equally acceptable to use a separate table (with a different name) to store temporary relationships.[10]

The problem is that surrogate keys are normally restricted to use within a specific, limited domain (e.g., within a single computer or a set of computers [e.g., within a single agency]). What is required is a surrogate key management system that is universally unique, so that no matter where the data it identifies propagate, it is always guaranteed to remain unique. In a military context, this may occur across services, coalition partner, and civilian boundaries. A strategy for accomplishing the goal of *universal surrogate keys* (USKs) is presented later in this section.

1.2.2 Organization-to-Organization Relationships (OTOR) and Roles

Most entities in a database are somehow associated with each other. As illustrated in Figure 4, every command relationship (i.e., link) between organizations has a set of properties associated with it. These command relationships are coined *organization-to-organization relationships* (OTORs) and include the command relationship type, its purpose, and any temporal constraints of the link. Many OTOR types already exist with names such as *assigned*, *attached*, and *operational control.* A short list of some common OTOR types is presented in Table 2. Appendix A contains a more complete list of OTOR types and their current definitions from various sources.

Table 2: Common DoD and NATO OTOR Types

From the DoD Dictionary:	
Assigned	*Administrative Control (ADCON)*
Organic	*Direct Support*
Attached	*Reinforcing*
OPCON (Operational Control)	*General Support*
TACON (Tactical Control)	*In Support of*
From the NATO Dictionary:	*Operational Command (OPCOM)*

[10] For example, a new table with the name "Temporary Attachments" or "Task Organization" could be used for this purpose.

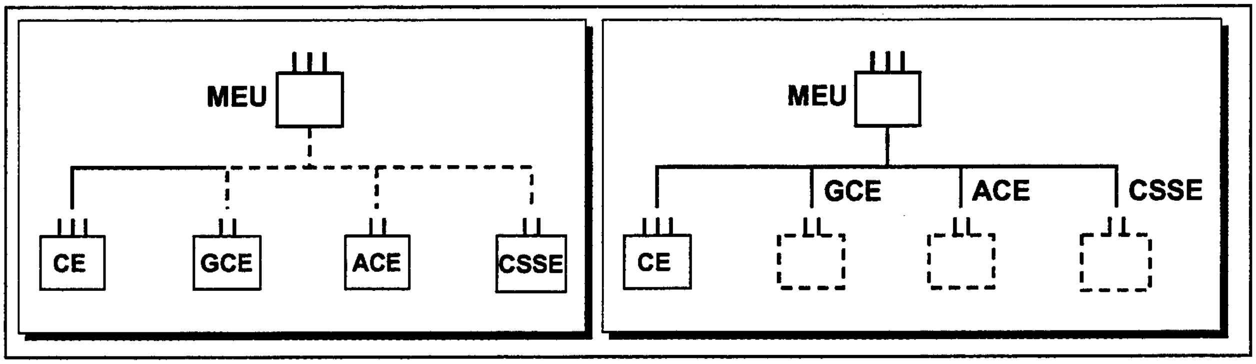


Figure 5: Temporary Nodes Versus Permanent Links.

Similar in appearance to OTOR types but different in purpose are *roles*. Informally, a role is descriptive attribute, normally a doctrinal *title*, which describes the reason for a particular relationship. In other words, a role is a named link, not a node. Figure 5 illustrates the concept of roles. The left organization chart is a common depiction of the composition of a marine expeditionary unit (MEU) that has four basic elements: a permanent command element (CE), and attached ground and air combat and combat service support elements (GCE, ACE, and CSSE, respectively). Their temporary, attached status (OTOR) is depicted by dashed lines. In reality, this is not an accurate representation of the situation. There are no real organizations in the U.S. Marine Corps with the official titles GCE, ACE, or CSSE. Rather, these are *roles* (i.e., links) that existing organizations accept. For example, a U.S. Marine Corps rifle battalion is augmented (task organized) with additional organizations (such as amphibious assault vehicle units) and is given the temporary alias of specific battalion landing team (BLT), which in turn, is attached to the MEU as the GCE; the BLT is serving the role of GCE. A pictorial version of this representation is illustrated in the right organization chart of Figure 5.

Because every deployed MEU has a GCE, ACE, and CSSE, these links can be permanently reserved ahead of time for these habitual roles. This is illustrated in Table 3 where the *Child* column is left blank until deployment time when a real unit's identifier will be inserted for each of the habitual roles. The advantage of this approach is that links (with stable *Org-Assoc-IDs*) can be predefined and treated as reference materiel, and the same association instance can be reused for each deployment. This allows "well-known links" to be defined so that other databases know exactly where to find this information without costly queries or network searches. The advantage of roles is further demonstrated in the section describing crews.

Table 3: Database Tables Showing Named Links or "Roles."

NODES

Organization Table	
Org-ID	Name
2349	"22d MEU"
409	"22d Cmd Element"

LINKS

Organization-Association Table					
Org-Assoc-ID	Child	Parent	Assoc Type	DTG	Assoc Name
22132	409	2349	Assign	-	Cmd Element
7773	(TBD)	2349	Attach	-	Ground Element
22399	(TBD)	2349	Attach	-	Air Element
74242	(TBD)	2349	Attach	-	Combat Service Support Element

1.2.3 Allocation of Organization Identifiers (Org-IDs)

Recall that organizations are defined as the nodes of a graph that compose a unit. It is recommended that a surrogate key be used as the primary key for organizations per the example in Figure 4 (Chamberlain 1998). Further, it is recommended that a simple 4-byte (32-bit) integer be used and this primary key attribute be named *Organization Identifier* (Org-ID). Each organization (or node) receives a universally unique Org-ID, thus allowing command structures to be explicitly defined via parent-child relationships between organizations (as illustrated in Figure 4). A 32-bit integer has the capability to uniquely enumerate almost 4.3 billion organizations (actually, 2^{32} or 4,294,967,296), provided that all the possible numbers are used (i.e., none are wasted).[11]

In the past, human beings have had a strong tendency to permanently subdivide or allocate address space ahead of time. This is purely to assist in human interpretation and intervention; the machines do not care. Therefore, the second part of the recommendation is to treat the surrogate keys as they were intended, merely as integers, with no special assignment of values to the bits (i.e., no assignment of patterns for specific purposes, such as if bit x and y are 0s, then it is an Air Force organization). Integer values should be assigned first come, first served, with no waste. This practice has three major advantages: (a) it allows the computers to do simple integer operations that are very fast because all references to organizations are handled as integers; (b) it provides a very terse and efficient manner to identify organizations; and (c) it is very general and prevents humans from interfering by encoding information into the bits, which they will later learn to regret. By using an approach conducive to machine manipulation, one can build a complete tree structure simply by including a single integer attribute in each organization, which references its parent organization (be it the default, current, or some other definition of parent). This enables very quick access up and down the organization tree using traditional tree traversal operations (tree graph data structure algorithms).

1.3 The Organization Server Architecture

Military organizations and units are world-wide entities. Therefore, implementation of a system to assign unique surrogate keys to organizations should be designed to be flexible enough to operate across military service, non-governmental, and international boundaries. This requires a universal way to identify organizations to be defined with a capability to provide organizational information about military forces, in a homogeneous structure, down to any arbitrary level desired, including the individual warrior or billet level.[12] As previously mentioned, to prevent waste, Org-IDs must *not* be assigned by blocks but on a first-come, first-served basis. Therefore, it is recommended that implementation be via a two-tier hierarchy of servers: (1) a limited set of *Org-ID servers* to distribute unique Org-IDs and (2) an arbitrary number of *organization servers* (org servers) to maintain the assignment of Org-IDs to organizations and the associated voluminous organizational data. This approach is illustrated in Figure 6.

---

[11] 4.3 billion numbers are enough to uniquely identify every node in more than 200,000 U.S. Army divisions.

[12] Dictionary of Military Terms - http://www.dtic.mil/doctrine/jel/doddict/data/b/00875.html
Billet: A personnel position or assignment that may be filled by one person

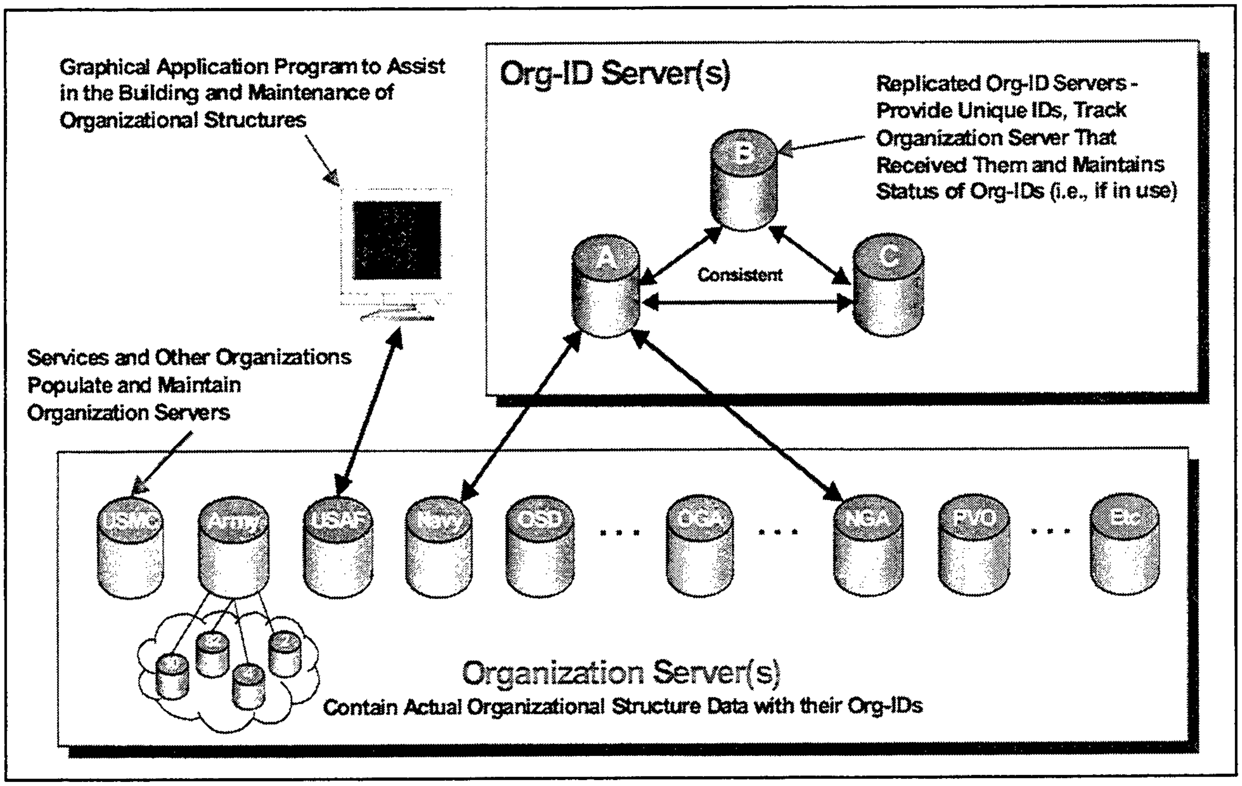


Figure 6: Organization Server Architecture

1.3.1 Organization Servers Versus Org-ID Servers

The separation of Org-ID servers from org servers is important because it allows selective sharing of OOB information and participation in the open, Org-ID allocation process. It is anticipated that this will encourage participation by those who would not otherwise do so for security or sovereignty reasons. The jurisdiction of org servers is completely flexible. For example, within the United States, there could be org servers for each DoD service, the Office of the Secretary of Defense (OSD), other Government organizations (e.g., State Department), non-government organizations, and private voluntary organizations. Each org server controls the dissemination of its information, and it is expected that DoD members would normally share information with each other (within normal security limitations). Further, any of the org servers may be *virtual* in the sense that physically, they are actually several dispersed machines. In Figure 6, this is illustrated with the Army organization server that could be composed of org servers at lower echelon organizations (e.g., Army divisions).

Org-ID servers interact only with each other and org servers. Individual applications are not allowed to communicate directly with an Org-ID server. The org servers are controlled and maintained by the force development agencies of the various organizations (e.g., the U.S. armed services). These agencies have the responsibility for developing the force structure for their service and ultimately provide the staffing and other documents that define how the services are organized, populated, and equipped. In other words, org servers contain administrative information about the *default* structure of organizations, and populating these servers is an administrative function conducted by the force development community. In the next section, exactly what this implies is explained in detail.

When an authorized force developer wants to create a new organization, an Org-ID (or set of Org-IDs) must be obtained. To do this, an organization server sends a request to one of several Org-ID servers. The Org-ID server authenticates the request and provides guaranteed unique integers to the requester, that is, the Org-ID server ensures that the integers it provides have not been given to any other authorized user. This is illustrated in Figure 7. For each Org-ID allocated, the Org-ID server associates the org server to which it was delivered and a status. The status of a newly allocated Org-ID is designated as *dormant* (i.e., unavailable) until the org server reports that the Org-ID has been assigned to a real organization, thereby activating it. This will probably be required within some previously agreed upon limit to prevent hoarding of Org-IDs. An org server may inactivate an Org-ID at any time, reuse it as it pleases, or even return it to the Org-ID server for reuse. The Org-ID server has minimal control over an Org-ID once it is delivered to an org server.

Operational users (e.g., military units in the field, such as a joint task force staff) of organizational data interact only with org servers and never with the Org-ID servers. As operational units pass unit task order (UTO) information via their operations orders (e.g., the temporary links illustrated in Figure 4), it is possible that the data may include unknown Org-IDs if all the required default organizational data have not been previously disseminated (which should be a rare event). When default organizational information is required, it is requested and obtained through the org servers. This information could be about an organization (i.e., a single node) or about an entire unit (a root node and all the default descendants down to some level).

Normally, all organizational information required for a military operation is pre-loaded into computers during the planning process (or is already residing there). However, it is possible that an unknown Org-ID could be encountered during operations. To learn the identity of unknown Org-IDs, a request is sent to any org server. This is illustrated in Figure 8. If the org server has the

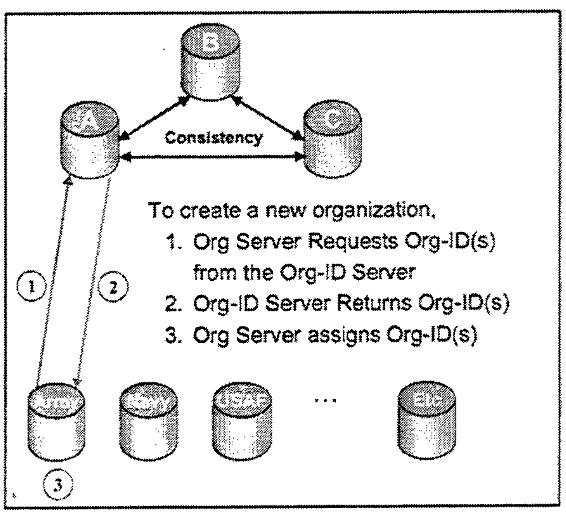


Figure 7: Creating a New Organization

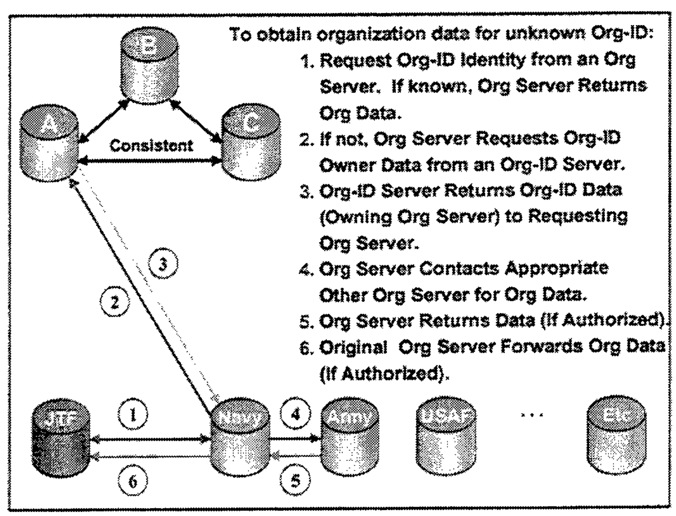


Figure 8: Identifying Unknown Org-IDs

information locally available, it returns the information to the requestor. If the information is not locally available, then the org server must request it from the Org-ID server to learn whether the Org-ID is active, and if so, to which org server the request must be directed. Once this information is returned to the org server, several options are available. Figure 8 illustrates the case when the original org server retrieves the final organizational information and forwards it to the requesting user. This is the desired approach if there is limited bandwidth between the requestor and the org server. It is expected that there will be high speed network access between org servers and Org-ID servers because these assets reside in secure, administrative areas. However, if the network link to the user is equivalent to that between the various servers, then it is equally valid to pass the identification of the responsible org server received from the Org-ID server directly to the requestor. The requestor would then contact the appropriate org server for the organizational information. The point to be emphasized is that org servers provide services to users, while Org-ID servers provide services to org servers.

Handling requests for organizational information that is not local to an org server raises basic security policy issues. Each org server controls the dissemination of its organizational information. Therefore, each org server must maintain a list of authorized users. The policy issues concern trust that other bona fide organization servers will correctly enforce the access limitations. For example, in the previous example (illustrated in Figure 8), the implications are that the Army server (1) shares its access constraints with the Navy server and (2) trusts the Navy server to properly enforce its access privileges. If this can be accomplished, it simplifies the access process because *any* org server can handle a request for information from any user, thus allowing authentication between a user and an org server to be established only once. However, if the mutual trust approach cannot be agreed upon, then a less efficient but more compartmentalized approach can be implemented, which requires independent org server access authentication at a cost of reduced efficiency, flexibility, and robustness. It is anticipated that a sound engineering solution can be readily attained via commercial off-the-shelf (COTS) products for cases when a policy is agreed upon.

1.3.2 Org-ID Stability

Although the reader may at this point have a vision of a huge bottleneck being created as thousands of users flood the org servers to create new organizations for doing battle, this is not the case. The basic property of Org-ID *stability* has a significant impact on implementation feasibility. This concept is illustrated in Figure 9. The top half of the figure depicts two pure Army companies; on the left is a pure tank company and on the right is a pure mechanized infantry company. Both companies have three platoons (each depicted with three dots over the box symbol). The tank company contains only tanks (e.g., M1A1 Abrams main battle tanks) and the mechanized infantry company contains only infantry fighting vehicles (e.g., M2A2 Bradley fighting vehicles).

To obtain a more flexible and robust fighting force, it is customary to mix tanks and infantry fighting vehicles. This is demonstrated in the bottom half of the figure where the "3rd Platoons" from each company have been switched, as is indicated by the dashed line. Observe that no new nodes were created; there is a total of ten nodes in both the pure and task organized versions. However, the graphs have changed. In the task organized version, two new (temporary) links have

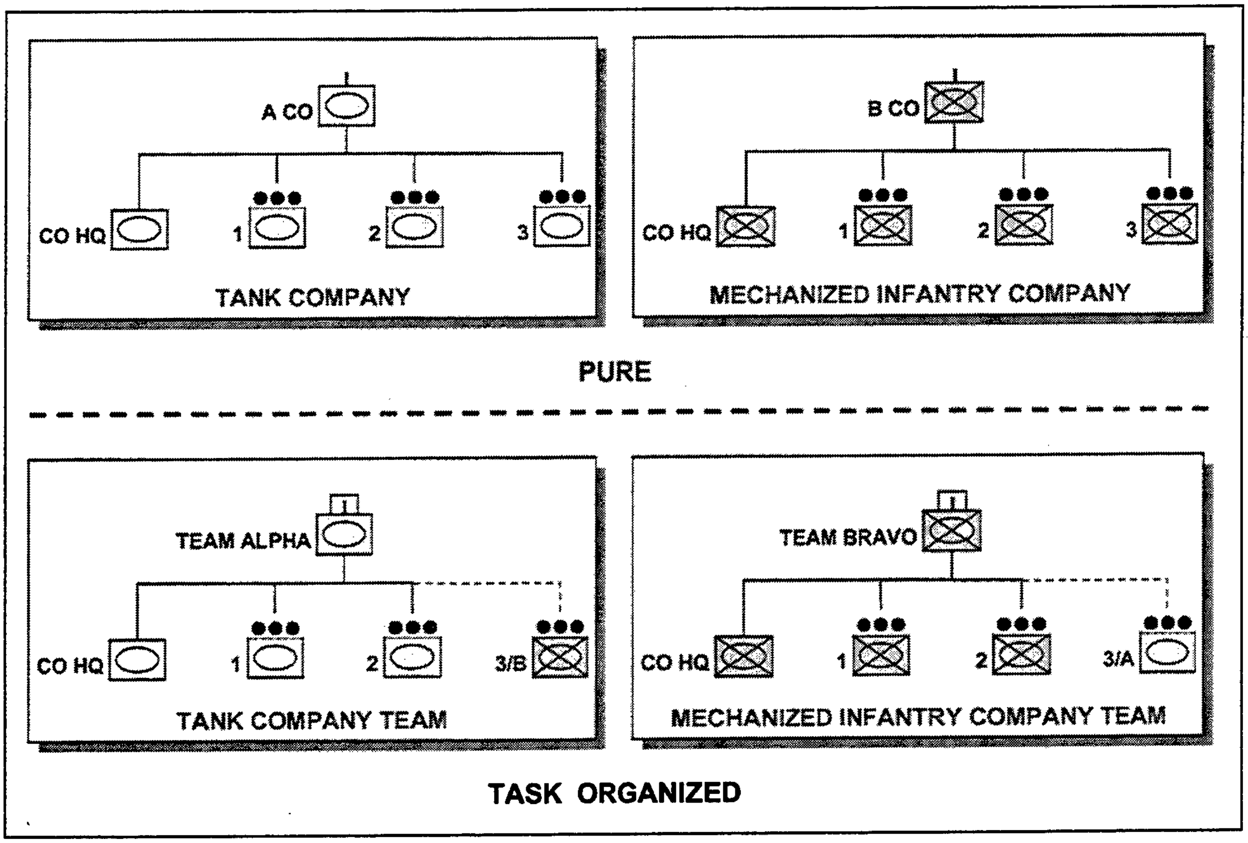


Figure 9: Task Organizing Does Not Require the Creation of New Organizations

been added that associate the "3rd Platoons" with a new parent even though they still retain their default relationship (i.e., assigned) with their original parent.

Organizations are assigned an Org-ID when they are created, which they keep for life. This means that in most cases, building a new task organization does not require the generation of new Org-IDs (i.e., creating new nodes in the organizational graph). Instead, links are added to temporarily restructure the existing organizations. In other words, the identities of the individual nodes are static, while the links between them are dynamic. Only in rare cases, such as the creation of a joint task force in which a new organization is truly created, should an ad hoc visit to the Org-ID server be required. Pre-allocating a few Org-IDs to the unified commands for contingency operations can circumvent these situations, however.

To visually identify this situation, several other features have been included. First, the parent organizations may receive a temporary alias. In this example, A Company becomes Team Alpha and B Company becomes Team Bravo, although any name may be assigned by the planner. Second, the echelon symbol obtains a "hat" to notify observers that it is no longer a pure entity. The point emphasized is that although names, links, and symbols have been temporarily modified, *no new organizations were created* by definition. Only new links were added, or in formal terms, the graphs were modified. This is the key point of this report.

Obviously, this does not occur only in the Army. In a Navy example, a specific destroyer squadron, two submarines, and a supply ship are temporarily attached to a core carrier or cruiser destroyer group to build a carrier battle group. In a Marine Corps example, a battalion landing team is constructed and is temporarily attached to an MEU. In an Air Force example, a strike package is created by combining several existing aircraft, each represented by an organization. Therefore, dynamic and fluid operational command structures are created by re-linking a stable set of "functionally static" organizations that have fixed Org-IDs. The assignment of Org-IDs to this stable command structure is an administrative function performed by the force development agencies of the services. The trick is to produce a default command structure that mimics the way we fight, and this command structure, called the *default operational organization* (DOO), is explained in the next section.

2 DEFAULT OPERATIONAL ORGANIZATIONS

If one asks 10 different people to "draw an organization chart" of their organization, one will often receive 10 different structures. This characteristic is indicative of the informal, human-oriented nature of our battle command processing. Unfortunately, computers are not clever enough to discern what all these relationships imply. Consequently, one needs to formally describe the familiar command structure relationships that are routinely used and have numerous implied meanings and ramifications.

One reason for establishing the org server system is to simplify, enhance, and formalize the task organization process within and between U.S. services and coalition partners. Two basic hypotheses were presented earlier (and appear in Chamberlain 1999), which are based upon the formal definitions for organization, command structure, chain of command, and unit:

Hypothesis 1: A default command structure exists that is relatively stable and if designed carefully, can be used as the base structure for integrating entities and building arbitrary *orders of battle*. This is called the *default operational organization (DOO).*

Hypothesis 2: Operational command structures are fluid and are nearly always constructed by modifying (i.e., re-linking the nodes of) DOOs.

Corollary 2a: When building an operational command structure (OOB), one should rarely have to create new organizations. However, one may want to provide an alias for an existing organization.

Therefore, *OOBs* can be created and disseminated by exchanging lists of command structure links (i.e., using pairs of Org-IDs) that identify an organization, its new parents, the time and duration of the temporary association between them, and other attributes required to fully describe the relationship. For this to be true, however, nodes must exist to which to link. In other words, the default structure must reflect the way the unit fights (or deploys), and it must include more nodes than are present in current default command structures. Just entering current staffing documents into the org servers will not accomplish the desired capability because many important, especially low echelon, operational organizations are merely implied or are completely missing.[13] Therefore, these organizations must be explicitly added to allow the org server system to meet its potential.

The term DOO is introduced as the name for the augmented organizational structure to accomplish this goal. The DOO is the conceptual "stable" generic organization structure that represents the "pure" state of the force before it is task organized. It begins at some arbitrary high echelon (e.g., DoD) and continues all the way to the individual warriors. DOOs are maintained in the org servers.

A DOO is constructed in two phases. First, the organization, represented within standard staffing documents, is translated into "nodes" and "links" (i.e., organizations and a default command structure). This may require some basic modifications to remove circular, duplicative, and ambiguous links that can cause logic problems and prevent the graph from being a tree. Second, depending on the features of the particular staffing document, three other types of organizations are added to fill the many gaps left after translating the staffing documents. These organization types are *billets* (organizations that have a one-to-one correspondence with a person), *crews* (organizations that are created for the purpose of operating a piece of equipment), and *doctrinal organizations* (organizations based upon operational fighting doctrine or heuristics). Billets will be leaf (terminal) nodes, while crews and doctrinal organizations will be internal (non-terminal) nodes.[14] While the identification of crews is relatively uncomplicated, selecting doctrinal organizations that can add many internal nodes is more complex. The next sections describe these cases and some anomalies discovered during their definition.

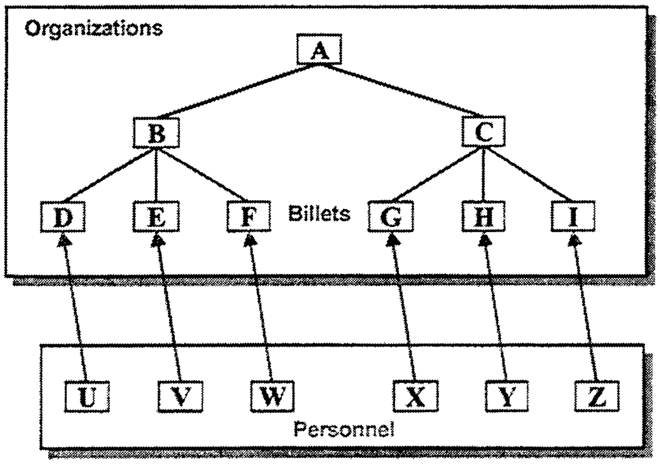
2.1 Billets and Leaders

As one descends the organizational hierarchy, eventually one encounters organizations that have a one-to-one correspondence with an individual person (called *billets* in the official DoD dictionary). Billets are the "leaf nodes" of the organizational tree, and approximately 80% of the organizations in a force are billets (see Appendix B for an explanation of this statement). Currently, tracking of military organization extends to the "company level" in the Army and Marine Corps, "squadron" in aviation units, and the ship in the Navy. Units at these echelons are assigned a unit identification code (UIC) for strategic planning and logistical purposes. However, UICs were established for logistical reasons, not for command and control. The current echelons were chosen because they represent supply delivery locations. It is important to realize that this is an arbitrary stopping point in the process of defining organizations. Figure 10 illustrates this relationship between billet organizations and personnel.

In the general case, billets correspond to "warrior" positions. These are the personnel positions that are occupied by soldiers, sailors, airmen, and marines doing the many tasks required of a military force. People are ultimately assigned to billets, but this does not change the fact that billets are simply another organization.

[13] Staffing documents include Army *Tables of Organization & Equipment* (TO&E), Air Force *Unit Manning Documents* (UMD), Navy *Ship Manning Documents* (SMD), and Marine Corps *Tables of Organization / Table of Equipment* (TO/TE).

[14] Recall that leaf nodes do not have children while internal nodes do.



Obviously, a person is not a billet, but this is easy to forget.

General Case:

Ultimately, the leaf nodes of an organization chart (a tree) represent the individual warriors of the organization.

A "billet" (or "personnel slot") *is an organization* that has a 1:1 correspondence with a person. (Nodes D - I correspond to billets.)

People get assigned to billets.

By treating billets as just another organization, personnel and unit databases can be integrated together.

Figure 10: Billets Ultimately Form the Leaves of the Trees

However, the association between a person and a billet is technically different than the association between two organizations. To complement OTORs (organization-to-organization relationships), which were introduced in Section 1.2.2, *personnel-to-organization* relationships (PTORs) link people to organizations or more specifically, to billets. In Figure 10, the links between organizations (such as A and B or B and F) are OTORs, while the arrows between people and organizations (such as U to D or Z to I) are PTORs.

While many OTORs are already well established (e.g., *assigned*, *attached*, *operational control*, and *direct support*), PTORs are less so. Both the number and description of OTORs and PTORs may have to be modified to formally address the many cases that occur in real life. One example of a situation requiring a special PTOR is representing the practice of fulfilling the role of another person who is temporarily absent (e.g., a company commander on leave). In this case, a temporary PTOR is added to indicate that a person occupying one billet is also temporarily serving in the capacity of another. Another interesting situation, not unusual at the upper echelons, is the assignment of a person to several billets at one time. This is referred to as "wearing several hats." For example, the person filling the billet of Commander, U.S. Forces Korea (USFK, a sub-unified command of the U.S. Pacific Command), also fills the billet of Commander, United Nations Command (UNC, formed under U.N. charter), and Commander, Republic of Korea-U.S. Combined Forces Command (CFC, formed by a bilateral agreement between the U.S. and R.O.K.). The opposite may also be true; a billet may be occupied by more than one person. This can occur when an over-strength condition exists or when the skill mix of the warriors does not match the skill mix authorized for the unit. These cases and many others must be formally addressed if a computer system is to be used to represent and manipulate information, based on conditions that arise from these practices.

2.1.1 Leadership Billets

A special case is the "leadership" billet. It is important to be able to ascertain who is the leader of an organization (or unit). However, leadership billets are often nested several layers down from the organization that is actually led. For example, the leadership billet for an Army division is a sub-element of a "Command Section," that is a sub-element of a "Headquarters and Headquarters Company" (HHC), that is a sub-element of the organization called a "division." This relationship is relatively easy to understand, given some military experience coupled with an organization chart that includes echelon symbols and organization names as in Figure 11.

However, a more formal description is required for computers that must address an abstract case such as that shown in Figure 12. To fix this problem, a new OTOR is introduced called *is_led_by*, which links a leadership billet with the organization that it leads. This is illustrated in Figure 12 between nodes F and A and between nodes H and B. (Note: Figure 12 is the abstract case of the example shown in Figure 11.) Because a PTOR exists to assign person X (flesh and blood) to organization F (a leadership billet), it is easy to determine that person X is the leader of organization A (e.g., MG Jones is the leader of the 32d Division in Figure 11). In formal terms,

$$\textit{is\_assigned\_to}(X,F) \ \textbf{AND} \ \textit{is\_led\_by}(A,F) \underset{\text{(implies)}}{\longrightarrow} \textit{is\_leader\_of}(X,A),$$

which is a derived PTOR. Thus, no matter where a leadership billet is placed, one can easily ascertain who is in charge of an organization.

Notice that the *is_led_by* link goes from an internal node to the billet. Although this is somewhat arbitrary, it is preferred to the reverse since an internal node is only allowed to be led by one billet, while a billet may be the leader of several internal nodes. For example, when an Army tank platoon leader is in his tank, fighting the platoon, the platoon leader billet serves as the leader of at least three organizations (internal nodes): (1) a tank platoon, (2) a tank section (pair of tanks), and (3) the tank crew. It is simpler to envision each of these three organizations as having a single reference to a leader than having a leader with three separate references to organizations. However, either technique is equally viable.

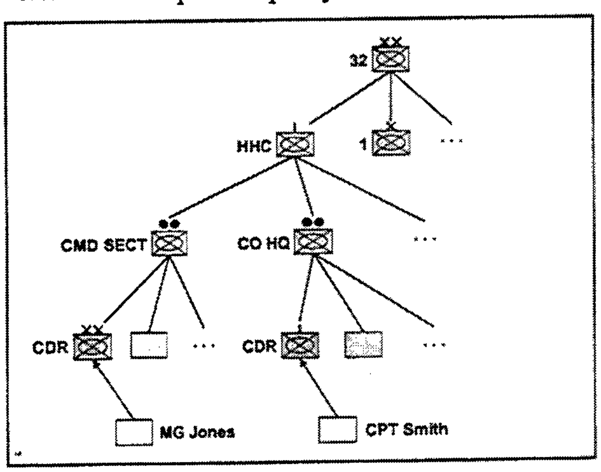


Figure 11: Nested Leadership Billets

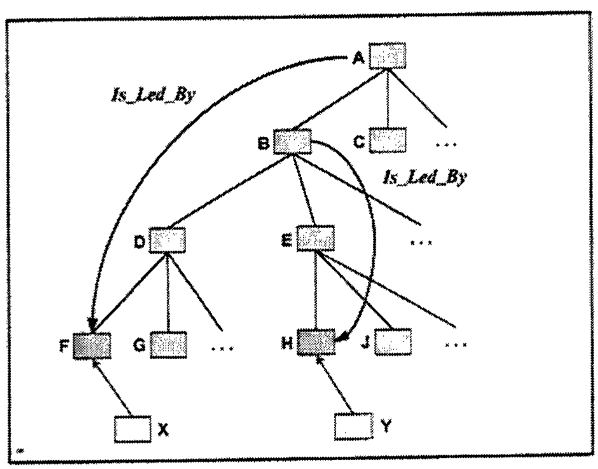


Figure 12: Abstract View of Nested Leadership Billets

2.1.2 Chain of Command Versus Command Structure

A chain of command is different from a command structure. A *command structure* is a tree graph that depicts the decomposition of a unit into its subparts. This is the basic formalism used to describe unit composition, and because a command structure contains internal nodes between a root organization and its billets, it is useful for aggregating information about descendant nodes. A *chain of command* is also a tree graph, but it depicts a "leader-subordinate" relationship between billets and therefore contains ONLY billets, not internal nodes. The links of a command structure are read as *is_composed_of,* while the links of a chain of command tree graph are read as *reports_to*.

A chain of command can be automatically derived from a command structure, provided that the *is_led_by* relationships are included. However, the reverse is not true since an infinite number of command structures can be constructed from any given chain of command. Figure 13 and Figure 14 illustrate the similarities and differences between these two common tree graphs. In this case, Figure 14 is the chain of command derived from Figure 13. Notice that the structure in Figure 13 is composed only of billets. Many times, when units are being designed, it is preferable to view a chain of command chart rather than a command structure chart. Fortunately, a simple automated operation can switch between the two views.

The algorithm for constructing a chain of command from a command structure is provided in Appendix C. This algorithm appears more complex than expected because it handles several anomalies, such as internal nodes not having *is_led_by* links. A basic tenet being considered is whether all internal nodes should be associated with an *is_led_by* link. The rationale is that every group of warriors has, by decree, a de facto leader (at least in the U.S. military). However, there may be organizations without any billets under them, so in these special cases, there may be either (1) leaf

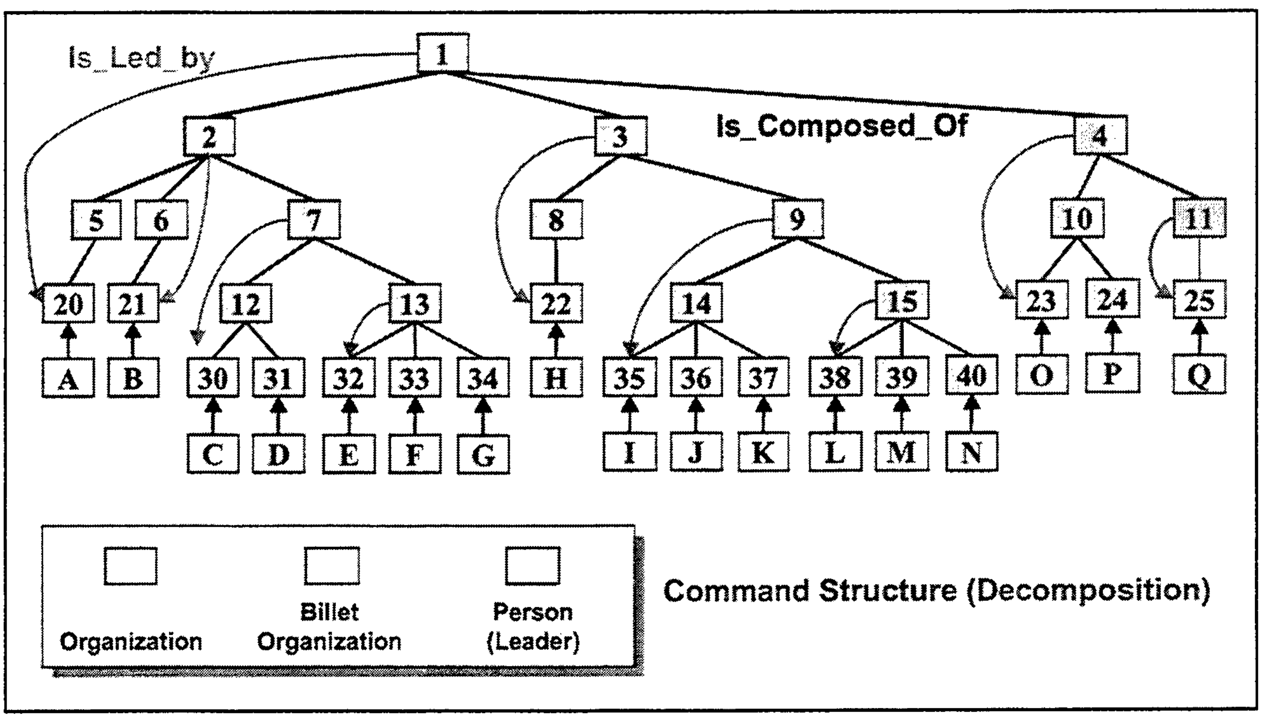


Figure 13: A Command Structure Depicts Decomposition

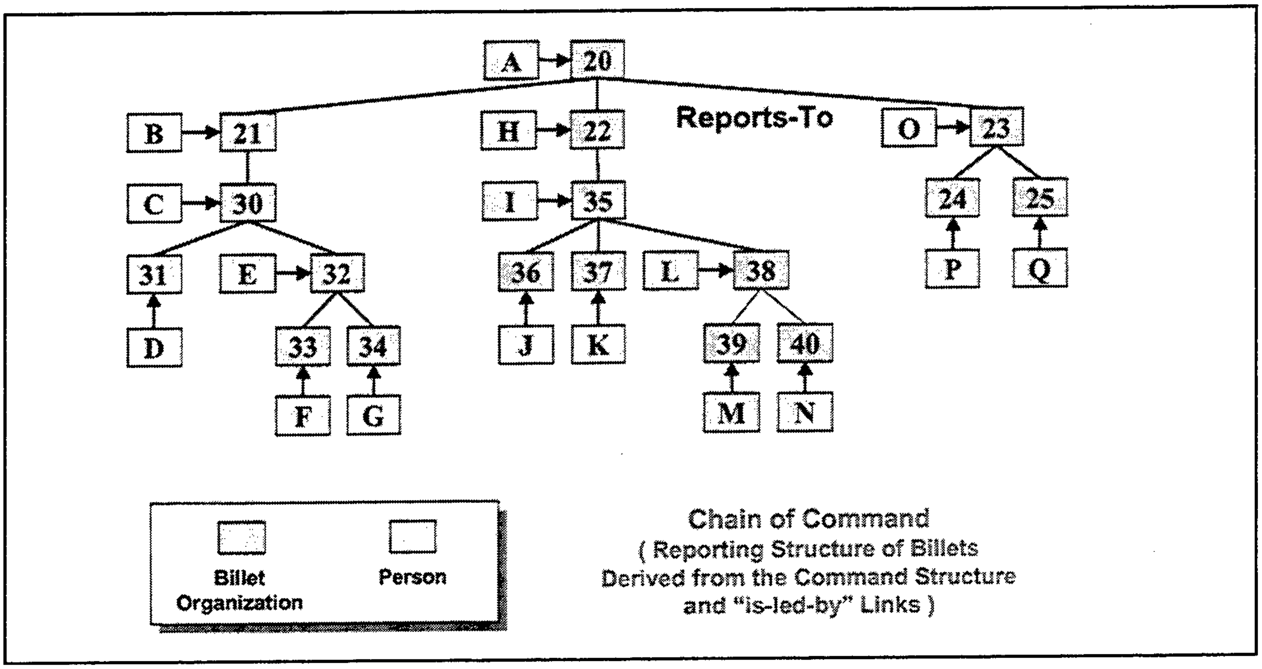


Figure 14: Chain of Command Derived From Command Structure

nodes that are not billets or (2) nodes without a leader. This concept is still undergoing scrutiny, but a good argument against it (for cases when leaf nodes are billets) has yet to be discovered.

In spite of all these links, an ambiguity still exists. Because Army tables of organization and equipment (TO&Es) are logistically based, their tree structures are focused on logistic responsibility and materiel ownership rather than on command composition. Notice that in Figure 12 (and Figure 11) there are two leadership billets under one node (i.e., nodes F and H under B, which, in turn, is under A). (Note: This situation is illustrated in Figure 13 with nodes 20 and 21 being under node 2, which, in turn, is under node 1.) This produces an ambiguity when rules of transitive closure for trees are followed.[15] A basic characteristic of a leadership function should be that if B *is_led_*by A, and C *is_led_by* B, then, ultimately, C *is_led_by* A. In other words, if one is the leader of an organization, then one should also be the leader of *all* those organizations that are part of it. There is a contradiction in Figure 12, however, in which both nodes F and H are below node B (and in Figure 13, where nodes 20 and 21 are below node 2). Using graph terminology and Figure 12, notice that the result of converting the command structure to a chain of command differs, based upon which node is selected as the root (beginning) node. If one begins the conversion using the tree rooted at node A, the following relationship is derived:

CASE 1:
[A *is_composed_of* B] AND [H *is_a_descendant_of* B] AND [Y *is_assigned_to* H];
Further, [A *is_led_by* F] AND [X *is_assigned_to* F];
THEREFORE Person X must be a leader of Person Y.

In other words, Person X *is_a_leader_of* Person Y.

[15] Transitive closure states that IF A=B AND B=C, THEN A=C. Likewise, the '=' relationship may be replaced by some other relationships: IF A *is_an_ancestor_of* B AND B *is_an_ancestor_of* C, THEN A *is_an_ancestor_of* C.

However, if one begins the conversion using the tree rooted at node B, a different conclusion results:

CASE 2:

Like H, [F *is_a_descendant_of* B];

Further, [B *is_led_by* H] AND [Y *is_assigned_to* H] AND [X *is_assigned_to* F],

THEREFORE Person Y *is_a_leader_of* Person X.

Both of these cases cannot be true.

The result of the algorithm should not depend on which node (level of the hierarchy) one begins the process. This situation is caused by an anomaly in the graph that is a created when a billet is both a descendant and a leader of an intermediate organization that has a leader. All descendants of an organization should be governed by the leader of that organization. So it is a contradiction to say that one organization is both a leader and descendant of another organization.

For example, in current, logistically based Army staffing documents, a division commander billet (that is authorized a Major General) is part of a Division Command Section that is part of an HHC that also has its own leader (a company command billet is authorized an Army Captain). This practice occurs at all levels of the Army above the company echelon (where HHCs exist) because the HHC commander has responsibility for the leader's equipment. Thus, the practice of building "official organization charts" based on logistic responsibility rather than on chain of command causes command anomalies if not treated correctly.

The solution for this problem is straightforward. Any organization that is not subordinate to an ancestor is moved from under the ancestor and made an equal to it. This is accomplished at the highest aggregate level above the "higher ranking" subordinate. In this example, illustrated in Figure 15, the command section (CMD SECT) is moved from under the HHC to become an equal sibling to it. This revives the "tree property" of the organization chart and provides a clean command structure because all the nodes under the HHC are now led by the leader of the HHC. However, to reflect the practice that the HHC organization is responsible for the equipment that is used by its superior organization's command section, an OTOR called *has_logistics_responsibility_for* is added to formally specify this requirement.

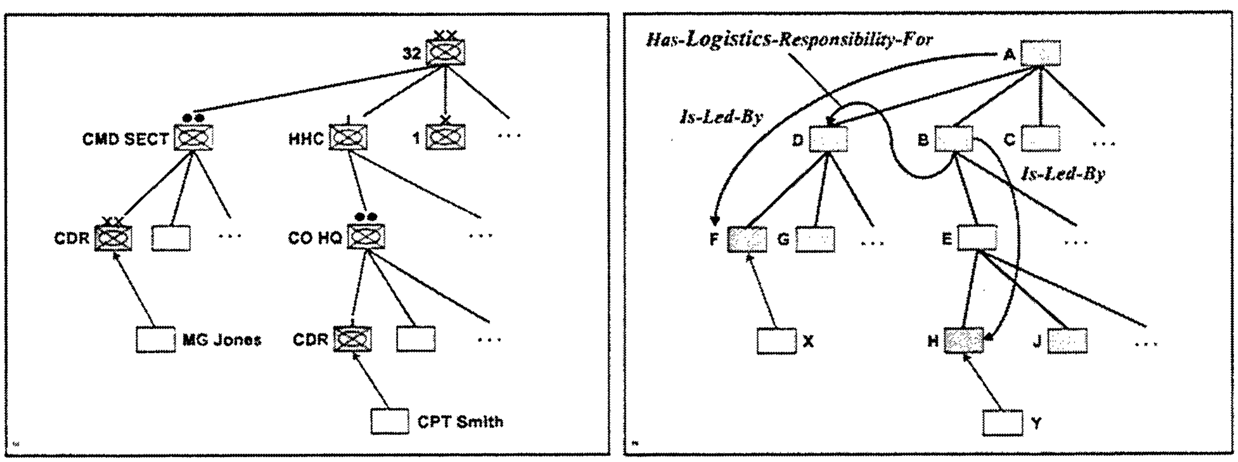


Figure 15: Non-nested Command Structure

This discussion of billets and commanders illustrates how common practices of the past that passed scrutiny by knowledgeable humans are not always appropriate for automated techniques. On some occasions, this is because computing power is lacking or because of excessive algorithm complexity required to manipulate data in ways akin to humans. However, in other cases, such as the command section anomaly just described, it is attributable to a long-held misinterpretation of what is intended by the document at hand. In this case, expanding the use of staffing documents from a predominant logistic focus to a wider, operational focus requires some common practices to be modified to reflect the new intended purpose. Sometimes, relatively minor structural changes can have a significant impact on the overall system and current business practices.

2.2 Crews

It is a common practice in all the military services to create organizations, commonly called "*crews*," for the purpose of operating a specific piece of equipment. Crew sizes vary widely from a single person, such as a fighter pilot, to thousands of people, such as a ship's crew that is further organized into several echelons of sub-organizations. The purpose of this section is to identify common traits and practices that can be applied to the representation of crews, regardless of size, complexity, or the type of equipment that is operated. For the remainder of this report, the term "asset" is used interchangeably with "equipment."

Formally, a crew is an organization, with an Org-ID, whose purpose is to tie together people and equipment using one common entity. Obviously, there are many ways to do this, so it is advantageous to provide some guidelines, or semantics, about how to compose crews with a variety of available associations. Of particular interest are the associations between (1) the individual crew members and (2) the crew and their asset. For clarity, two terms are defined to differentiate between these two associations:

Alignment: the association between an asset and an organization.
Assignment: the association between the organizations (e.g., billets) of a crew.

Composing a crew requires both assignment and alignment. To formally represent alignments, a new category of relationships, called *equipment-to-organization* relationships (ETORs), is introduced to accompany OTOR and PTOR. Using these definitions, assignment uses OTORs, while alignment uses ETORs. Thus, the associations between organizations, people, and equipment can be explicitly defined using OTORs, PTORs, and ETORs. This is illustrated in Figure 16.

There is a famous saying that "the Air Force and Navy man equipment, while the Army and Marines equip the man." Although this is an obvious exaggeration, it reflects a philosophical difference in the center of attention of the crew composition process. The two primary distinctions are whether (1) the process is *centered* on crew assignment or crew alignment and (2) the associations involved are *habitual* or *non-habitual*. These characteristics have generated an historical divergence in the structural composition of units that include crews. The goal of this discussion is to formally describe these apparent differences and to demonstrate, using the graph terminology already introduced, that they define two equally valid approaches for representing

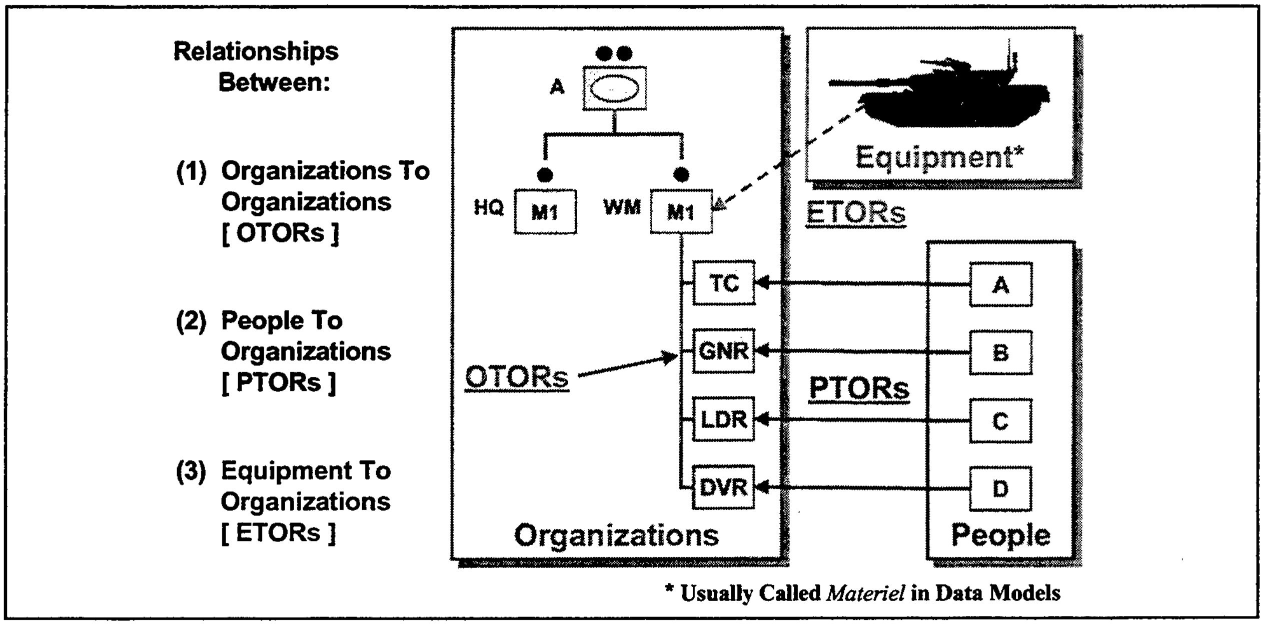


Figure 16: Three Categories of Relationships: OTORs, PTORs, and ETORs

similar organizational constructs.[16] Fortunately, these differences are illusory and can be explained in a unified, coherent scheme.

Many interesting ramifications are caused by the distinction between habitual and non-habitual relationships. All associations may have this attribute, whether it is an OTOR, PTOR, or ETOR. An habitual association is one that has a predefined, recurring connection between entities. An example of an habitual assignment is a crew whose composition is based upon a *specific* set of billets. In such cases, the reason the billets exist is because they are part of a specific crew. To the contrary, non-habitual associations do not have predefined bindings. Although the entities involved may have an established association that is based upon a set of documented operational procedures, the binding between the entities is not established until it is required for operation.

Although an association may be either habitual or non-habitual, PTORs are nearly always habitual. Typically, a person is assigned to a billet for the duration of a tour of duty. However, this is not so for crew composition, as is illustrated in Figure 17.

For ground and naval forces, both crew assignments and alignments are typically habitual. Habitual crews obtain a sense of identity because they continuously work with the same people and operate the same equipment. It is common for habitual crews to consider an asset "theirs" ("my vehicle" or "my ship"), and they perform basic maintenance on it. Force designers in this domain typically create crews, based upon a ratio of *one crew to one asset*, and the organization charts reflect this perspective because the crews are often composed of other organic organizations (often billets).

Aviation units, on the other hand, are typically non-habitual in alignment and assignment. The crew rarely expects to get a particular asset (i.e., aircraft). Aircraft are provided, based upon

[16] To quote a corollary of Murphy's Law: "Where one stands [on an issue] depends on where one sits."

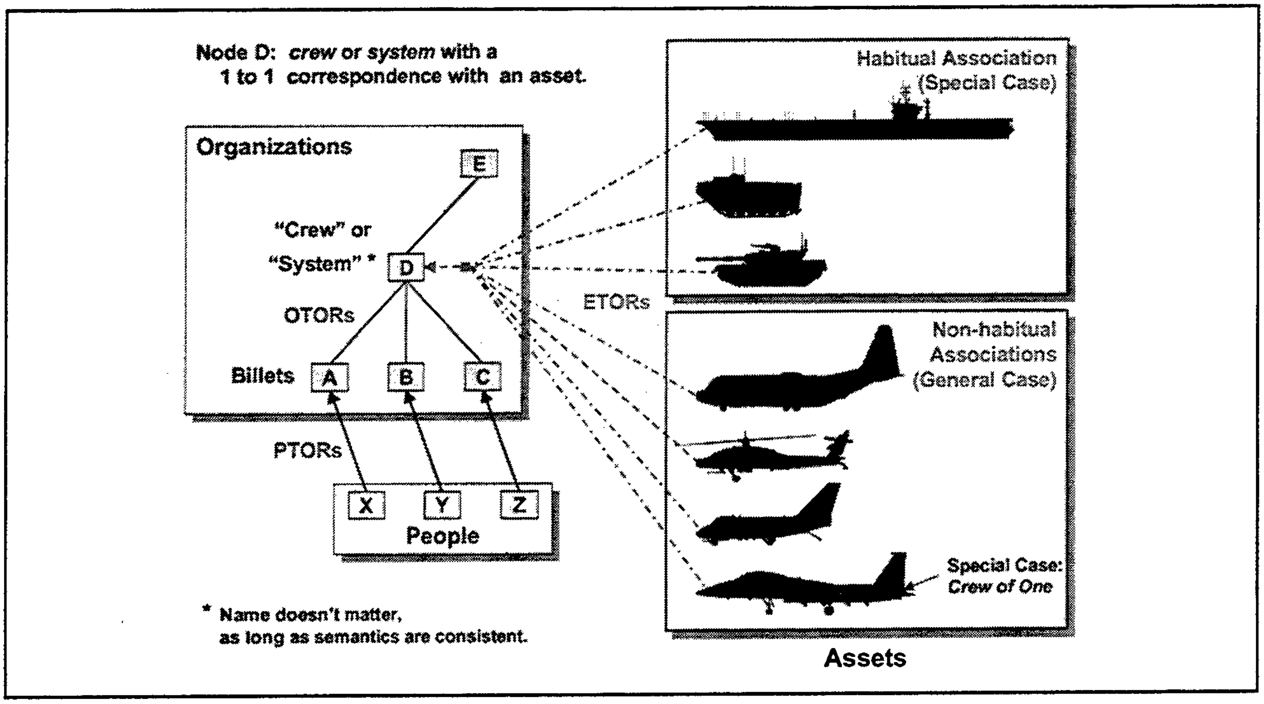


Figure 17: Example of Habitual and Non-Habitual Relationships

maintenance and other scheduling criteria; no one has a "personal" aircraft.[17] Although an attempt may be made to keep crew members together, this is by operational practice rather than by permanent organizational structure. Consequently, force developers in this domain typically base organizational structure and size on ratios of qualified crew members to number of assets. The organization charts reflect this perspective since crew members are usually completely separated from the organizations that are aligned with the assets. As a result of these practices, the term "system" usually seems more accurate than "crew" as a description of the organizations that are aligned with assets. This is perfectly acceptable as long as the semantics are consistent.

Intrinsically, crew organizations are no different than other organizations except for the fact that they correspond to a major asset, such as a vehicle, aircraft, or ship. Crews follow the same composition and construction semantics as other organizations. Both habitual and non-habitual relationships are equally viable. Neither is predominant over the other, and any semantics must be general enough to handle both cases. Because the habitual case is more constraining, however, the non-habitual case should be considered the general case, and not vice versa. Figure 18 illustrates four combinations of crew and equipment relationships.

Quadrants 1 and 4 describe the two common (extreme) cases of habitual and non-habitual associations; typical examples of these are ground fighting vehicle crews (Quadrant 1) and aircraft crews (Quadrant 4). Quadrant 3 represents the case when a crew always works together but operates different pieces of equipment, based on the operational conditions. Examples of this exist in special operations forces (SOF) where an habitual team uses whatever asset they need to complete a mission. Although this is the ideal case for aircraft crews, if it is achieved, it is attributable to operational

[17] People in organizations that are habitually aligned are surprised to discover that the pilot of a fighter aircraft is rarely the person whose name is on the aircraft!

policies and not to the organizational structure; therefore, it does not fit in this category. Quadrant 2 represents the case when the members of a crew always work on the same piece of equipment, but there is no fixed crew composition. This condition is common in "shift work," such as maintenance crews and watches. Any semantics for organizational structure must be capable of representing all four of these conditions.

	Assignment (Team Membership)	
Alignment (Equipment Used)	**Habitual**	**Non-Habitual**
Habitual	1 **Same Team Members And Same Equipment**	2 **Same Equipment But Not Same Team Members**
Non-Habitual	**Same Team Members But Not Same Equipment** 3	**Ad Hoc Equipment And Ad Hoc Team Members** 4

Figure 18: Four Combinations of Crew Relationships

2.2.1 Assignments, OTORs, and Organizational Trees

Recall that the term "assignment" is used to refer to crew organizational composition. Habitual crew assignments are the normal default for ground and naval ship forces. For example, an U.S. Army M1A1 tank crew is composed of four specific billets: a tank commander (TC), a gunner (GNR), a loader (LDR), and a driver (DVR). The four people assigned to these billets consider themselves as a permanent team. Operationally, unless there is a significant anomaly, they train and fight together as a team. When they "fall out for formation" in the morning, they will be standing next to each other; they may even be roommates.

In organization charts, habitual crew assignments manifest themselves by the billets being displayed under the crew *as the default case*, as is illustrated in Figure 19. Using general OTOR terminology, the billets are *assigned* to the crew. This does not imply that they cannot be *attached* to other crews but only that in the resting state, they are considered to be permanently part of the crew. In this example, the basic force development process entails creating one crew for each asset that is required by the parent organization. Then, based upon the specific asset involved, a set of specific billets is created and assigned to the crew. Thus, the organizational structure is based upon (1) a one-to-one correspondence between a crew and an asset and (2) a template of the crew composition. This same approach applies equally to large crews (as documented in ship manning documents [SMDs] in which the subordinate organizations include both internal nodes and billets.

Although habitual crew assignments are desirable, they are not always practicable. Aviation crews (when greater than one pilot) may be composed "ad hoc" based upon many issues (e.g., a crew member's need for training hours).[18] Although an attempt may be made to keep crew members together, it is not reflected in the organization charts because this association is based on an independent operational policy that is implemented on a best effort basis.

Figure 20 illustrates the situation, as just described, that is common in most aviation units. Note that, contrary to Figure 19, the flight crew billets are not organized into predefined teams. Instead, the flight crew billets reside under some other organization. For example, in an Air Force squadron, they typically reside under a Flight; in a Navy squadron, they reside under the Operations Department. In the Army, they are placed under operational elements, such as platoons, and are named by the alternate duty they perform (e.g., in Figure 20, AMO is air maintenance officer).[19]

Under this structure, the flight crew billets are considered temporarily "attached" to a crew for the duration of a mission, after which, the billets return to their default (assigned) organizations. However, notice that the concepts of *predefined* command relationships or "roles" can be used to document the well-known crew positions (see Section 1.2.2). Therefore, to establish a crew, all that is required is to "fill in the blank" slots (for pilot and co-pilot in Figure 20) with the Org-ID of the billets filling the role.

Neither the habitual or non-habitual approach to assignments is right or wrong; they are just different. The important point to remember is that either case must be equally viable within a data model of organizational structure, and the user must be free to mix and match in any way deemed necessary to properly represent the structural state.

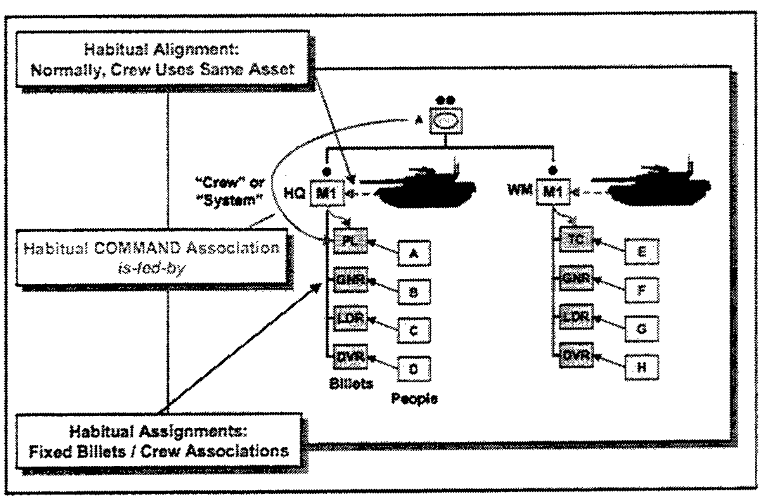


Figure 19: Habitual Assignments and Alignments

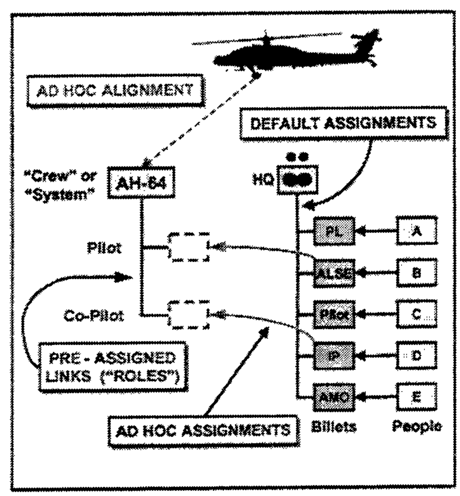


Figure 20: Non-Habitual Relationships

[18] An extreme example is airline crews, in which crew composition is determined according to seniority and other factors. Therefore, a pilot, copilot, and the several flight attendants may meet for the first time just minutes before a flight.

[19] This information is based upon official staffing documents: Air Force Unit Manning Documents, Navy Squadron Manpower Documents, and Army TO&Es.

2.2.2 Alignments, ETORs, and Organization Trees

The notable feature of a crew is the one-to-one correspondence with a major asset. This means that for every "crew-served" piece of equipment, there is a corresponding organization called a "crew" or "system." For example, there is a ship hull (a piece of materiel) called the "CVN-73." However, there is also a corresponding organization called the "Crew of the USS George Washington" (or some other name of choice) that is composed of departments, divisions, work centers, billets, and other subordinate organizations. If the piece of equipment named CVN-73 suddenly sank, there could still be a crew of $5{,}000^{+}$ sailors "floating in the water," complete with a hierarchical command structure. Thus, one should not confuse a piece of materiel with the corresponding organization. This is illustrated in Figure 17, where Node D is the corresponding crew organization for the various examples of materiel.[20]

Alignments (the association between equipment and an organization indicated via ETORs), like assignments, can be either habitual or non-habitual. Crews with habitual alignments always use the same asset (as illustrated in Figure 19), and as previously described, they consider the asset "theirs" (e.g., "my ship" or "my tank") and are often responsible for some level of maintenance. During the force development process, crew design is normally based on a one-to-one ratio of crews to assets. Consequently, the force structure that is deployed "to fight" (i.e., is task organized) looks very similar to the default force structure seen in garrison. For example, an Army M1 tank company has 14 tanks; therefore, 14 crews will be built into the force structure of the company. Whether in a deployed state or garrison, pairs of crews are defined as sections, pairs of sections are defined as platoons, and a set of platoons (usually three) constitute a company.[21] Although the tank company commander technically "owns" (i.e., has ultimate pecuniary responsibility for) all the vehicles, they are allotted on a semi-permanent basis to the leaders of each crew.

Non-habitual alignment is less constrained than habitual alignment and therefore should be considered the general case.[22] This is the normal default for aviation organizations. The primary reason for non-habitual alignment is that the assets (e.g., aircraft) require intensive maintenance on a regular basis, so it does not make sense to associate a particular crew with a particular piece of equipment. Because of the complexity of the systems, non-habitually aligned crews rarely do basic maintenance, and a full (often larger) maintenance organization usually coincides with the operational crews. During the force development process, the organizational design is normally based on a crew-to-asset ratio greater than one (e.g., 1.5 crew per asset) because the assets are (1) limited in number because of their cost and (2) complex to operate and require crew rest between missions. Thus, the force structure is designed to provide enough billets to keep the expensive assets fully employed during "around-the-clock" high paced operations.

As with the habitual case, the non-habitual case requires a crew organization to exist to aggregate people and equipment. At issue is the representation and inclusion into the overall organizational structure of the ad hoc crew organizations that have no crew members or assets

[20] Note that crews, like any other organization, may have several levels of subordinate organizations below them (e.g., a ship's crew), and ultimately, those subordinates will be billets. At the other extreme is an Air Force F-15C fighter squadron in which the asset requires only a single crew member (i.e., only one billet per crew).
[21] The other pair of tank crews operates as a section in the Company Headquarters.
[22] Force developers unfamiliar with non-habitual alignments often miss this fact.

assigned. There are two different perspectives from which to address this matter: an *asset-oriented* perspective or a *personnel-oriented* perspective. Either is equally acceptable, provided that the rules for manipulating the organizations remain consistent. As mentioned previously, it is perfectly acceptable to conceive of (and refer to) crew organizations as "systems" rather than "crews." Is it an M1A1 tank crew or M1A1 tank system (that includes crew members)? Is it an E-3C crew or E-3C system? Is it a DDG-51 crew or a DDG-51 system? The irony is that either approach is equally viable, yet a healthy debate can ensue over which is "the best." The simple fact is that there is no "best." Both approaches have their advantages and disadvantages that depend on the operational conditions and environment.

Habitually aligned crews are easy to define; one simply links an asset to the crew that uses it. However, with non-habitually aligned assets, the design of the organizational structure is not as simple. A consequence of a non-habitual alignment is a significant difference between the default and deployed (i.e., task organized) organizational structure. Since crew organizations are not habitually aligned with an asset, the primary issue is *how to account for and represent the assets organizationally when crews are not using them.*

The two basic perspectives are "system oriented" and "crew oriented." The system-oriented approach focuses on the asset itself and considers the organizational node a place holder for the asset. In other words, one thinks of the problem as "aligning people to assets" rather than "aligning assets to people." For example, each Air Force squadron has a number of prescribed authorized aircraft (PAA) that defines the number of aircraft allocated to the squadron. Using this approach, illustrated in Figure 21, there would be one "system" organization for each PAA. If the PAA were six, then there would be six system organizations called "PAA-1" through "PAA-6" (as an example). To each of these positions an aircraft (a piece of materiel) would be aligned via an ETOR. Since the asset is always aligned with a given PAA, this is equivalent to the relationships in Quadrant 2 of Figure 18 (i.e., the organization always aligns with the same asset, but a different crew is assigned each mission). The default structure might place the "PAA" organizations directly under the squadron root node because the squadron commander is ultimately responsible for all the aircraft.

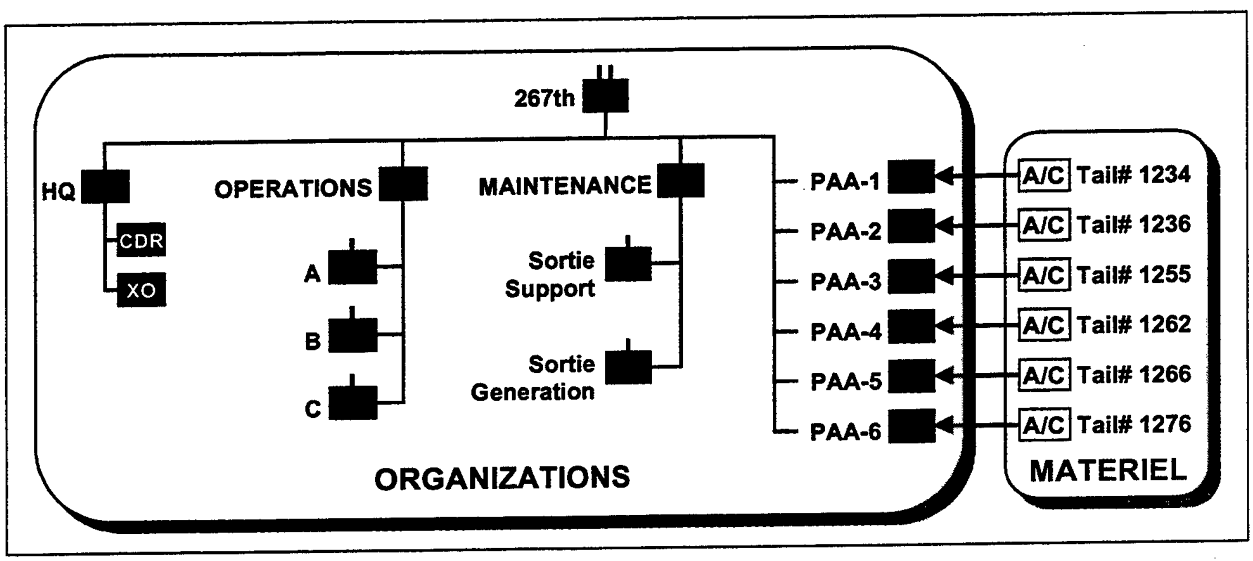


Figure 21: System-Oriented Approach to Non-Habitual Alignments

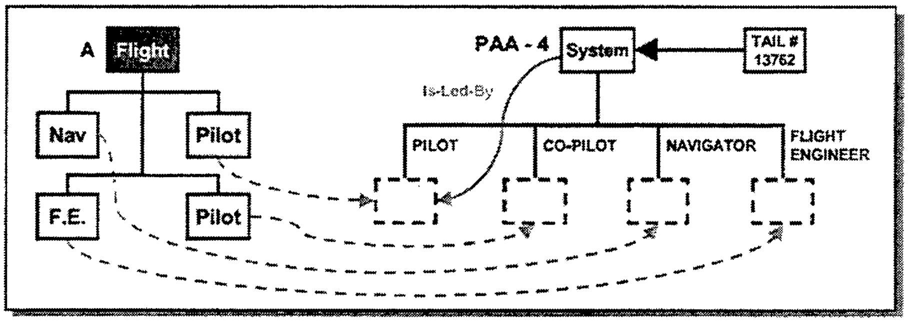


Figure 22: System Organization With Predefined Roles (links)

Each system node would have roles defined for the crew positions (e.g., pilot, co-pilot, navigator, flight engineer, etc.). When a mission is required, a flight crew is assembled by attaching billets, which reside under the operational flights (e.g., A, B, and C), to the predefined roles of the PAA nodes. This is illustrated in Figure 22.

One issue with this approach is the handling of maintenance crews (e.g., the aircraft crew chiefs). If crew chiefs are non-habitually aligned, then they can be treated the same way as flight crews by using predefined roles. However, if they are habitually aligned with an asset (which is often the case), then this complicates the process because this implies that they are assigned to the system. To prevent their being included automatically with the flight crew during a mission, they would have to be attached to another organization, thus causing a nuisance anomaly.

The opposite perspective is manifested in the crew-oriented approach. One may envision asset alignment as the establishment of a temporary ETOR with the organization that currently maintains responsibility for the asset. In other words, an asset "resides" with whatever organization is currently "signed for it." The major ramification of this approach to the unit structure is that there may be several "crews" for each asset. Although a one-to-one correspondence between an asset and a crew still exists, the reverse is not true. The overall number of crew organizations can be based on factors other than the number of assets.

Figure 23 illustrates a hypothetical example using an Air Force fighter squadron. In this case, two types of crews share responsibility for the aircraft: *flight* crews and *maintenance* crews. The aircraft is aligned with the crew that is currently responsible for it (illustrated with the dashed arrow). Initially, the aircraft resides with a maintenance organization that takes responsibility for it while it is not involved in flight operations. When the time comes for a mission, responsibility for the aircraft is transferred to the crew that executes the mission (led by the aircraft commander). When the mission is completed, responsibility is returned to the maintenance crew (led by a "crew chief"). This cycle continues, regardless of where the asset is located, and mimics the procedures used to sub-hand receipt items between responsible parties. The asset is always aligned to the organization responsible for it.[23]

One way to interpret Figure 23 is to consider the "system" organization of the previous perspective (e.g., PAA-4 in Figure 22) as being subsumed within the maintenance organization.

[23] Although this example has both crews within the same squadron, this is not the case with larger aircraft, such as the AWACS (airborne warning and control system), where separate squadrons (e.g., aircraft generation squadron) have responsibility for maintenance.

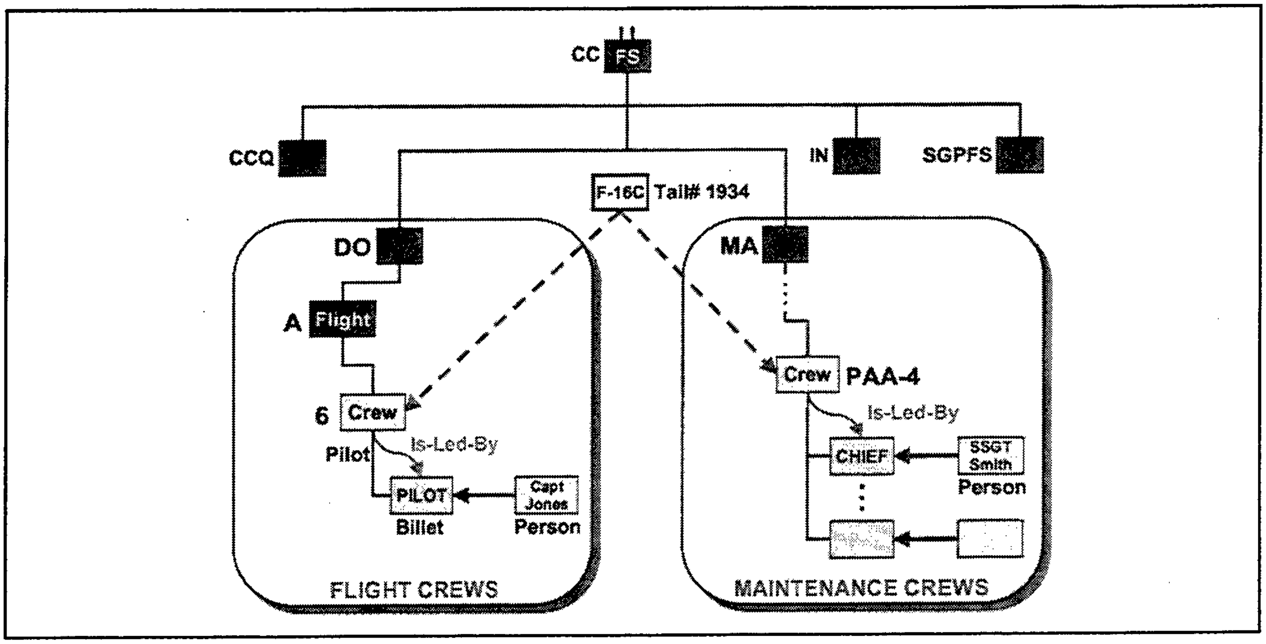


Figure 23: Crew-Oriented Approach to Non-Habitual Alignments

Then, a crew chief is assigned to it and the roles are removed because this organization is no longer used to aggregate members of a flight crew. Instead, the system node serves as the aggregation point for maintenance activities, and any billets created for this purpose may be added below it.

A new set of crews is added to serve as the aggregation point for flight crews under the squadron's operational flights. The flight crews may have default billets under them (as shown in Figure 23 for an F-16C crew that is habitual because of its singular composition) or may be completely separated from billets and have predefined roles as in Figure 22. In the latter case, the number of crews (as defined by the force developers) can be based upon a number of different factors. As a minimum, there must be one crew per asset. However, ratios larger than one crew per asset can be defined to support crew training and other activities that require automated tracking of crew attributes (e.g., on flight simulators or interactive war exercises).

Note the difference between the two perspectives. In the system approach, the alignment between the aircraft and a system organization is fixed and crew members are "attached to the system." In the crew approach, asset alignment is moved between organizations, based on which one has responsibility for the asset. Either approach is viable, as well as any mixture of the two extremes. However, in the next section (Doctrinal Organization), advantages of the second approach will become evident.

In summary, crew or system organizations serve as a central point by which to aggregate personnel with equipment, and they are pervasive throughout the armed services. This seemingly simple function has many facets. The associations between members and the alignment with their asset may be habitual or non-habitual. This characteristic results in a propensity to differentiate rather than integrate the concept of crews. However, both the crew and system perspectives represent a common theme whose semantics can be used across services and coalition boundaries.

2.3 Doctrinal Organization

Recall from graph definitions (see Figure 1) that every node in a tree is either a leaf node or an internal node. If a node has no successors, then it is a leaf node (such as a billet); otherwise, it is an internal node and has one or more successors. The number of successors of an internal node is called its *degree*. Figure 24 illustrates two extreme cases of trees, each with the same number of leaf nodes (i.e., 16). The top tree is has a single internal node (the root node) of degree 16. This represents the tree with the minimum number of internal nodes required to contain the 16 leaf nodes. The bottom tree is composed of 15 internal nodes, each with degree 2. This is called a *binary tree* and represents the symmetrical tree with the maximum number of nodes required to contain the tree. Observe that redundant nodes of degree 1 are disallowed.[24] For a binary tree, half of the nodes are leaf nodes, and half (minus one) are internal nodes. All other possible tree structures (of these 16 nodes) fall between these two extremes and have a smaller ratio of internal nodes to leaf nodes.

If the leaf nodes of an organization chart are billets, then the "art" of force development can be described as designing a tree structure with the right number and placement of internal nodes to support the mission of the force. If one wants to ensure that *any* combination of billets is possible, then a binary tree can be used for the force structure. However, most would consider this to be excessive, so less dense structures are typical. An approximate value for the degree of a symmetrical tree required to represent an Army division is provided in Appendix B. This results in an average degree of five (i.e., on the average, each node has five children), which means that 80% of the nodes in a division are billets and 20% are internal nodes.

The task of creating billet and crew organizations is an objective process. If one has a person or a major asset that requires people to operate it, then there is no choice but to create a billet or crew, respectively. However, the way in which billets and crews are clustered into super groups is a

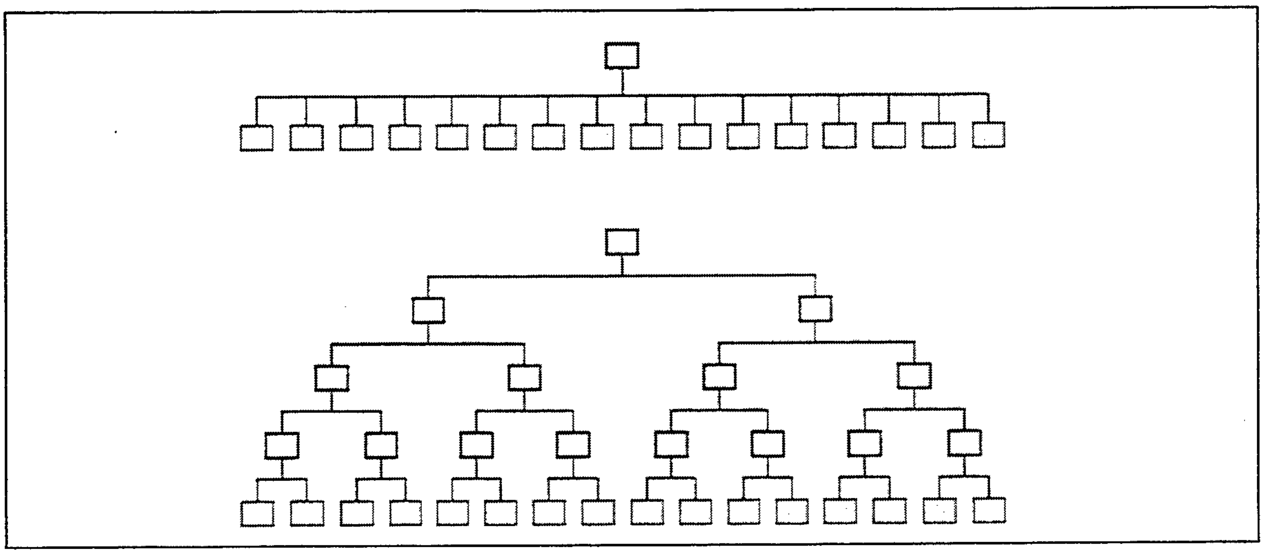


Figure 24: Two Extreme Cases of a Tree With 16 Leaf Nodes

[24] Normally, organizations of degree 1 are discouraged because they are redundant. However, crews are internal nodes and a special case is a crew of one, as is found in single crew fighter aircraft. This unique situation is tolerated to maintain consistency in the semantics of crew/system organizations.

subjective decision that is determined by operational concepts or *doctrine*.[25] For example, five soldiers comprise a fire team, two fire teams comprise a squad, and three squads comprise a platoon (although these numbers have varied slightly over the years). Similar variations have been common in all the services, whether one is discussing the composition of carrier battle groups, bomber squadrons, battalions, or ship crews. No single, general principle is used to guide the composition process of these units. The numbers of sub-units are derived by experts who base their decisions on weapon system requirements and on the associated tactics, techniques, and procedures (TTPs), or in general terms, the doctrine. Consequently, the many internal nodes that are used to group organizations into units (starting from billets or crews) have been coined ***Doctrinal Organizations*** and include Army divisions, Air Force wings, Navy fleets, Marine Corps expeditionary forces, and joint task forces.[26]

Doctrinal organizations serve as the place holders for information collected and aggregated from subordinate organizations, and they permit adjustments in information detail (or resolution) to propagate up and down the command structure. This is how war fighters track groups of people and equipment. If staffing documents are to provide a true representation of how we fight, and if they are to be effective in providing this information to battle command systems, then a comprehensive set of doctrinal organizations must be explicitly represented.

A problem that exists with current staffing documents is that they do not include many routine doctrinal organizations that are used for military operations. Instead, one must look in fighting and training manuals to find them. These unspecified doctrinal organizations are named *undocumented doctrinal organizations*, or ***UDO***s and span both administrative (i.e., default) and operational (i.e., task organized) organizational structures. UDOs usually occur or exist in the bottom three levels of organizational trees, directly above billets, and are commonly based upon operational heuristics. For example, the heuristic "always try to fight in pairs or greater" is almost universal; yet, in official organization charts, one rarely finds the organizations that are routinely employed as a result of this rule. From this basic heuristic, the concept of "lead" and "wingman" appear in every service's jargon as do the organizational entities of "section," "element," or "team." In the Army, Navy, and Marines, a pair of aircraft is often termed a "section," but in the Air Force, it is called an "element." These types of doctrinal organizations must be included if our force documents are to closely reflect the way we fight.

Within current staffing documents, a knowledgeable person can often infer UDOs by extracting information from other sources, such as billet titles. In practice, however, UDOs are discussed explicitly only in field manuals (FMs) and other documents that discuss tactics and operational procedures. Figure 25 illustrates this situation for a U.S. Army Mechanized Infantry Platoon. The leaves of the tree are billets and the four "M2 Crew" organizations are each aligned with a Bradley infantry fighting vehicle (BFV). Although platoons and squads (i.e., the top two levels of the tree) are described in the staffing document, there is no mention of fighting vehicle sections or fire teams (the uncolored organizations in the third level of the tree with the names A, B, or HQ). These UDOs

[25] *Doctrine*: a principle or position or a body of principles in a branch of knowledge or system of belief – Webster's collegiate Dictionary (10th Edition), 1996.

[26] The term doctrinal organization was coined by MAJ Mike Boller of the U.S. Army TPIO-ABCS, thus replacing previous, less descriptive terms.

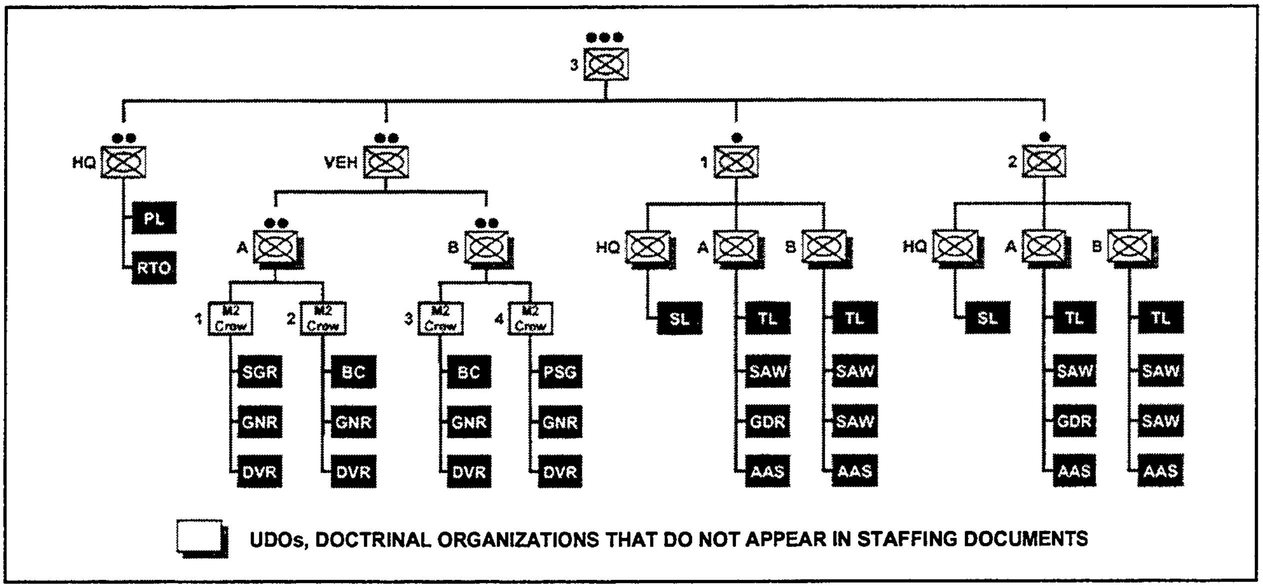


Figure 25: Doctrinal Organizations Missing From Staffing Documents

serve the important function of aggregating information about key tactical units and are essential in fully describing the structure of the platoon.

A frustrating chore associated with defining lower echelon doctrinal organizations is the inconsistent naming of the echelons. Figure 25 illustrates this situation because it reflects conflicting information between a staffing document and FMs.[27] The staffing document decomposes a platoon into four organizations: a platoon headquarters (HQ) section, a vehicle section, and two squads with the billets residing directly under these organizations. However, the FMs describe the platoon organization in operational terms and decompose it differently. In this description, the platoon is composed of two *elements*, mounted and dismounted, with the mounted element divided into two *sections (*A and B), each with two vehicles, and the dismounted element divided into two squads (first and second), each with two fire teams. However, the accompanying figure in the FM does not show the elements. Instead, it shows the platoon divided into five sub-organizations: a platoon headquarters section, two vehicle sections (A and B), and two squads (first and second).

The organization chart in Figure 25 is one attempt to merge these disparate descriptions—one of many possible composite structures. It ignores the elements but places the A and B BFV sections under the vehicle section, in essence, replacing the mounted element with the vehicle section. This example is indicative of the types of issues that force developers must address as staffing documents are augmented to reflect the way one fights or deploys. Minor issues such as "is it permissible to put a section under a section" and "must an official definition for the echelon called 'element' be established?" also surface.

Many names, such as element, section, team, detachment, and others are used sporadically throughout these levels. Unfortunately, unambiguous definitions for these organizations do not

[27] Information from: TO&E 07247F000, RFL CO INF BN (MECH) (XXI), Dated 23 Apr 1998
FM 71-1: http://155.217.58.58/cgi-bin/atdl.dll/fm/71-1/711-ch1f.htm#minfplt , section on Mechanized Infantry Platoon, and FM 7-7J, dated 7 May 93: http://155.217.58.58/cgi-bin/atdl.dll/fm/7-7j/Appa.htm#top

exist. However, an interesting outcome of using a formal tree structure for organization charts is that echelon names really become irrelevant to understanding the structure. From a computing perspective, it does not matter how one names the echelons, provided that one knows the *parent-child relationships*. In this context, the purpose of echelon names is relegated to supporting symbol drawing functions to portray units to the operators. In other words, it really does not matter what names are given as long as one knows who belongs to whom. If one maintains the standard symbol hierarchy, perhaps the BFV sections A and B should be marked as squads (using "single dot" echelon symbol) if this really matters.

However, the question remains whether there are any guiding precepts to help one decide when to insert a doctrinal organization into the official command structure. The answer appears to reside in the requirement to aggregate or track organizational information.

A basic tenet appears to be "if there is a recurring requirement to track a group of organizations, then there should be a doctrinal organization in place to do it." As an example, consider a hypothetical five-man infantry fire team. Suppose that there is a common practice of splitting the team into two groups (e.g., to serve as listening posts). Figure 26 illustrates three possible organization charts. Charts A and B show the two extremes; Chart A depicts the fire team with no subordinate organizations other than billets, while Chart B illustrates the case when two permanent doctrinal organizations are included to split the team into two symmetrical trees. Chart C portrays another option wherein the team leader is provided two "spare" organizations to use as he wishes. If there is a policy that the team leader always goes with one of the two groups (i.e., no one is left alone), then only one spare would be needed because the only splitting option is two teams, one with two members and one with three. The decision of which structure to use depends upon the requirements for aggregation and deployment.

A general characteristic of all *active* internal (non-leaf) nodes is that there should be a "boss" of the node. This means that every active doctrinal and crew organization should have an *is_led_by* link going to a billet to identify the person in charge. As mentioned in Section 2.1.2, this is required to build a chain of command from a command structure. The adjective *active* is significant because there may be doctrinal organizations without default bosses. Some organizations serve as place holders for habitual aggregation points and have periods of inactivity when a unit is not deployed; these are called ***intermittent organizations***. Command posts, crews of non-habitually aligned assets, and spare organizations are examples of intermittent organizations. These organizations may have significant dormant periods, but during deployment, they are essential for effective battle command

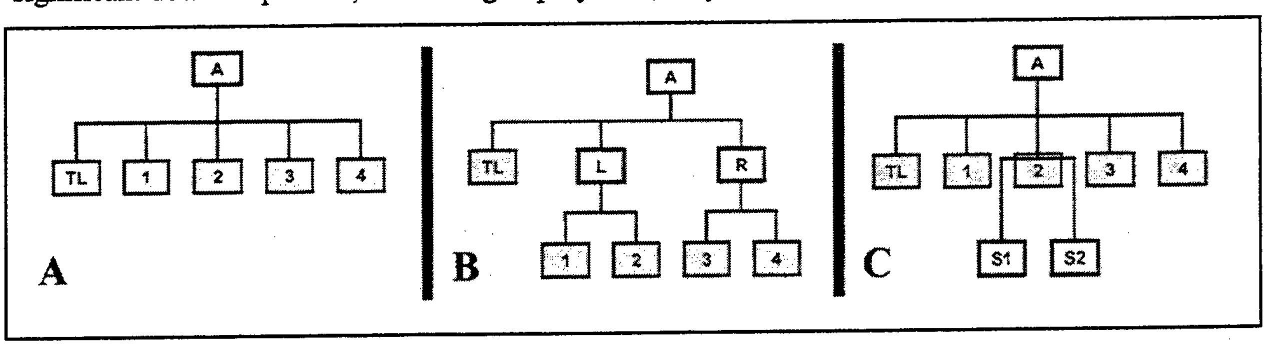


Figure 26: Example of Doctrinal Organizations

execution and monitoring. Therefore, to decide if an intermittent doctrinal organization is required, one should consider whether the necessary criterion (of having a boss) is met during the times when the intermittent organization is active. This requires that when an intermittent doctrinal organization is activated (i.e., when another organization is attached to it), a boss for it should be identified and an *is_led_by* link must be instantiated.

2.3.1 The Importance of Doctrinal Organizations

The importance of doctrinal organizations cannot be emphasized too much. Without a sufficient set of doctrinal organizations, a primary advantage of this process is not realized because *Corollary 2a* does not hold:

> ***Corollary 2a***: When building an operational command structure (OOB),
> one should rarely have to create new organizations.

For example, in Figure 26, Chart A has only one doctrinal organization available for linking.

Simply stated, to re-link arbitrary sets of organizations together, there must be something to which to re-link them. It is doctrinal (and crew) organizations that serve as the aggregation points for the re-linking. If they do not exist, new nodes must be created, and this will cause a burden on the command, administrative and communication systems of the units involved. For example, if one needs to track a pair of soldiers and a doctrinal organization does not exist to do this, then a new organization would have to be created. If this is accomplished via the organization server, substantial delays will be incurred, making this approach impracticable.[28] Thus, it is imperative that the time and expertise be committed to identify UDOs and to include them in staffing documents and other official force development documents. The primary aversion to this task is that identifying and adding doctrinal organizations to current staffing documents is a time-consuming and intellectually taxing activity that must be done by domain experts who have a rigorous understanding of military operations. However, the reward is great.

The significance of stable Org-IDs introduced in Section 1.3.2 can now be reiterated. If a comprehensive set of billets, crews (or systems), and doctrinal organizations is included in the official staffing documents, then all the organizations (nodes of the tree graph) required to build an OOB already exist; that is, they have been identified and given an Org-ID even if they are not always in use. When one builds an OOB, the existing nodes are re-linked by adding new, temporary command relationships between the nodes. Since, from a graph theory perspective, no new nodes have been created, it is argued that no new organizations have been created; instead, the command structures have been augmented, thus building a new unit (tree graph).

The beauty of this approach is that it means that a stable set of organizations can be created ahead of time to be re-linked at a later time at the discretion of the commander. In other words, although the links between organizations (OTORs), people and organizations (PTORs), or equipment and organizations (ETORs) are dynamic, the identification of the organization, via the Org-ID, is a stable entity. This has dramatic consequences because it allows the organizational database entities to become the stable rallying point by which all other dynamic attributes are tracked.

---

[28] A discussion of ad hoc organizations occurs later in this report.

For example, recall the simple task organization process illustrated in Figure 9. As a consequence of the temporary reorganization, the networked computing equipment of the organizations involved may require new network addresses resulting in additions or deletions from associated routing tables. All of this can be automatically reconfigured because the equipment, personnel, and communication channels are all related via the central concept of the organization (identified via Org-IDs). Thus, the formal organizational structure provides the enabling core information that is prerequisite for building truly adaptive and self-configuring systems.

2.4 Putting the Pieces Together

Three categories of organizations have been described: billets, crews (or systems), and doctrinal organizations. When all the pieces are combined, one has (by definition) a ***DOO***.

In other words,

Default Operational Organization (DOO) =

	Administrative organization (e.g., current TO&E, UMD, TO/TE, SMD)	(a)
+	Doctrinal organizations	(b)
+	Crews	(c)
+	Billets	(d)
+	Administrative organization "fixes" (e.g., Command Structure)	(e)

This is the organizational structure that is maintained in the service's organization servers, to be shared (to the extent desired) within the service and between joint and coalition partners. By including these organizational building blocks into one homogeneous structure, an immensely flexible capability is provided that allows commanders to construct any imaginable task organization with widely shared, predefined attributes and relationships.

2.4.1 Default Operational Organization Examples

The concepts surrounding DOOs are not new and have been used for years; they just have not been explicitly described. In this section, several examples are presented in the context and vernacular language used in this report. The first is an Air Force strike package and the second are Naval examples.

2.4.1.1 Aviation Example — A Strike Package

The first example is an Air Force strike package. Recall that the definitions of organization, command structure, unit, doctrinal organizations, crews, and billets were presented on pages 3 and 4. Figure 27 illustrates two organizations: on the left is a hypothetical slice of a DOO for an Air Force F-16 (single pilot) fighter squadron; on the right is a strike package composed from the DOO. Although this example represents a package built from a single organization, it should be clear that this approach could cut across arbitrary organizations, provided they subscribe to the same set of semantics.

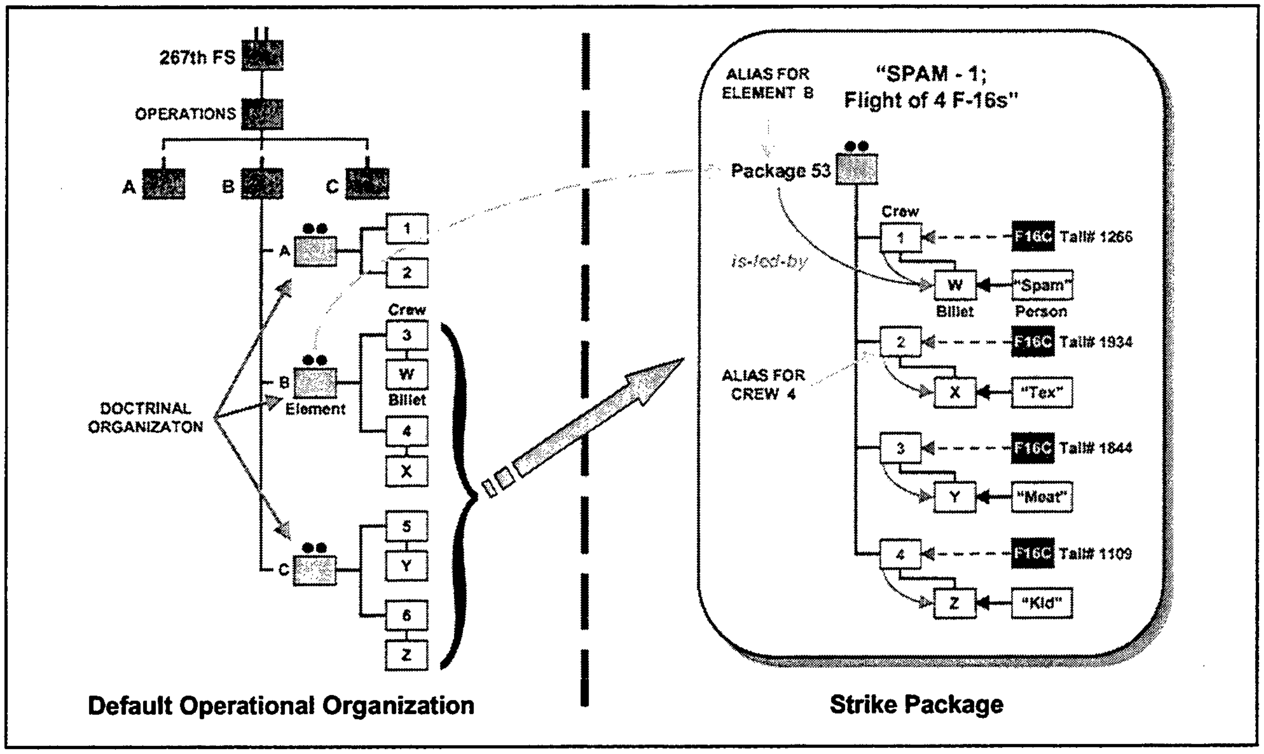


Figure 27: Building a Simple Strike Package

Normally, the official organizational structure stops at the flight level. Below the flight, tactical procedures are followed to combine aircraft into fighting elements, but they are rarely explicitly represented inside the organizational structure. As was previously explained, there are numerous options one can use to build a DOO; there is no correct, single solution. Recall that aviation units are typically non-habitually aligned and assigned (see Figure 18). In this example, the crew (versus system) approach is used because of the unique situation encountered with single-member crews (i.e., an F-16C has a single pilot; therefore, a distinctive one-to-one ratio exists between the crew member [a billet] and the crew organizations). To build the DOO, the flight is augmented with three echelons of organizations: elements, crews, and billets (pilots). Doctrinal organizations called *elements*, each composed of two crews, are explicitly placed under the flights.[29] One element is required for every two crews to ensure that any combination of aircraft can be grouped together when task organizations are built. If there are eight pilots in a flight, then four elements of two crews each would be required to provide total flexibility. The two-to-one ratio of crews to elements also reinforces the heuristic of always deploying in (at least) pairs of aircraft.

In this DOO example, the crew billets have been placed under the crew echelons, thus completing the organizational tree down to the individual person level. Since this example is an F-16C fighter squadron, there is only one billet per crew (in Figure 27, for simplicity, only the four billets involved and none of the people assigned to the billets are shown in the DOO). In this special case of single-member crews, there is a temptation to dismiss one of the two entities (crew or billet)

[29] ***Flight***: (DoD) 2. The basic tactical unit in the Air Force, consisting of four or more aircraft in two or more elements; http://www.dtic.mil/doctrine/jel/doddict/data/f/02472.html

as redundant, but this would deviate from the general nature of the problem. If application software expects billets under crews, then deleting crews (for example) would introduce anomalies. Finally, people (e.g., pilots) are assigned to the billets via PTORs and assets are aligned with the crew organizations via ETORs. In this example, it is assumed that the aircraft are aligned (via ETORs) with maintenance crews until the flight crews (i.e., pilots) use them.

When a mission is announced, a strike package can be constructed with only these predefined organizations; no new organizations need to be created. Conceptually, any combination of organizations may be temporary re-linked, although most are not tactically sensible. First, one of the element organizations of the squadron is selected to serve as the "strike package" root organization; in this case, it is Element B, and it is given a temporary alias during the operation (i.e., "Package 53"). Element B is chosen because one of its subordinate billets (i.e., Billet W under Crew 3) is occupied by the person selected to be the strike package leader (i.e., the person with nickname/call sign "Spam"). A temporary OTOR is added by an *is_led_by* link to explicitly identify him as the strike package commander.

Because Crew 4 is already a member of Element B, no new OTOR is required. However, two new temporary OTORs are required to attach Crews 5 and 6 to the strike package. The crews can also be given aliases to conform to standard operational practices. In this case, they are "renumbered" 1 through 4, or more precisely, "Spam-1" through "Spam-4." Finally, aircraft are aligned (via ETORs) to fill the package with the required assets. In this case, aircraft (materiel) with tail numbers are aligned with the crews of package 53. The final result is represented in the right half of Figure 27.

Figure 28 illustrates this same process when the system-oriented approach is used (rather than the crew-oriented approach). Although the initial DOO configuration is different, the resulting strike package structure is the same even though it is derived in a slightly different manner. Recall that with the system approach, there is one system organization per asset (e.g., PAA), rather than one crew organization per crew member set (see Figure 21 through Figure 23). In Figure 28, the billets reside under the flights, just as with the current Air Force unit manning documents (UMDs). For this example, the doctrinal organizations, the *elements*, are also placed there to keep them within the operations segment of the squadron; however, this is not a requirement.

To build a strike package, four steps are required whose order does not matter. In Step 1, just as with the crew-oriented approach, an element must be selected to serve as the strike package root and may be given a temporary alias for identification (e.g., "Package 53"). In Step 2, aircraft are selected for the mission and their corresponding system organizations are temporarily attached to the strike package root organization for the duration of the mission. In Step 3, pilots are attached to the predefined "pilot" roles of the system organizations. In Step 4, the package command structure is identified and the corresponding *is_led_by* links are inserted and the system aliases are assigned (e.g., Aircraft Crew 1 through 4).

From the war fighters' perspective, the resulting strike packages are identical, whether built from a crew-oriented or a system-oriented DOO. However, there are subtle differences. Recall that personnel allotment decisions for most are based on a crews-to-assets ratio that is greater than one to one; therefore, more crew (system) organizations are produced for a DOO with the crew-oriented

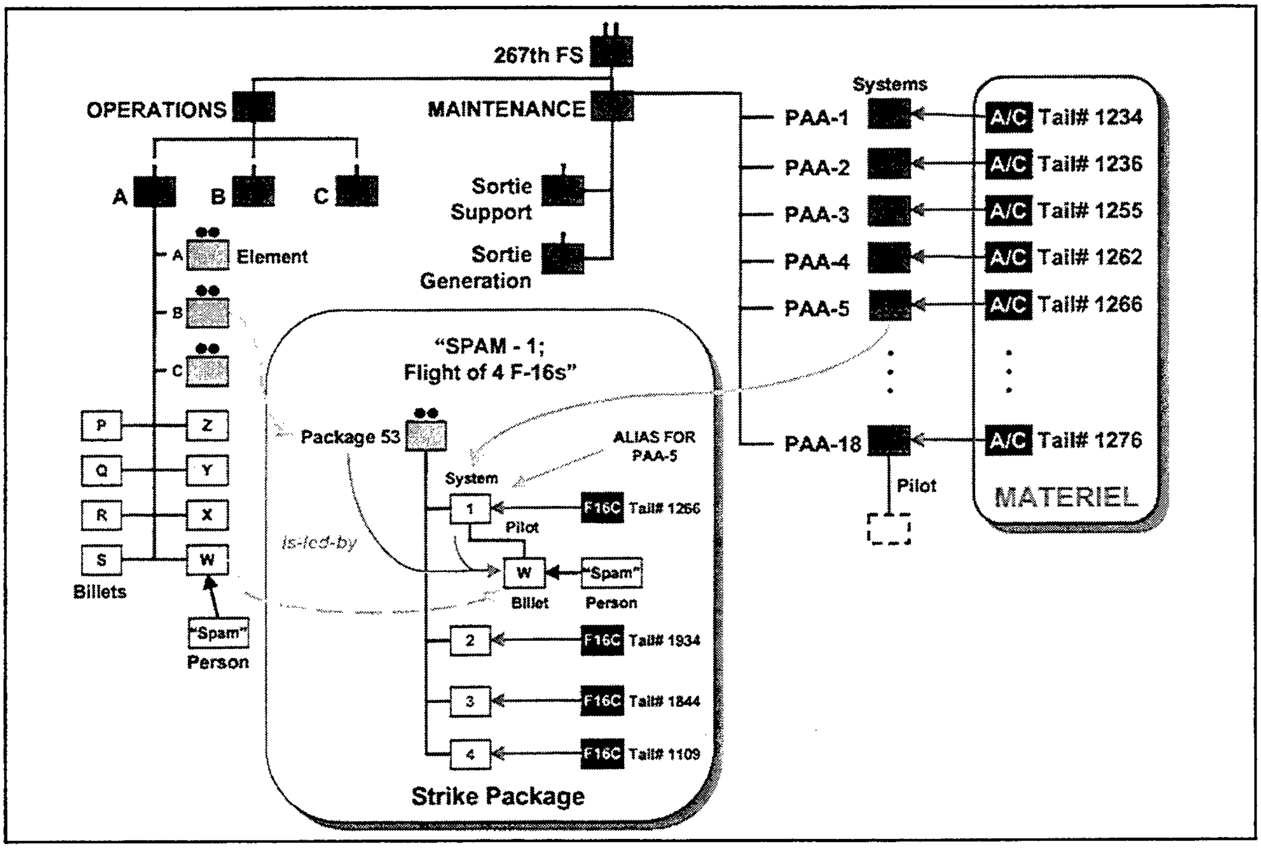


Figure 28: System-Based DOO for a Strike Package

approach than with the system-oriented approach. For these non-habitually aligned units, a benefit of the crew-oriented approach is that all crew members can be actively employed and accounted for as a crew, regardless of whether they are using an asset (i.e., flying a real mission). For example, suppose the PAA for a fighter squadron is 18 and there are 24 pilot billets to operate the 18 aircraft. Under the system-oriented approach, there would be 18 system organizations and if all 18 aircraft are flying missions, then all 18 system organizations are in use and there would be none for other purposes such as training in interactive flight simulators against other crews. Using the crew-oriented approach, there would be 24 crew organizations, one per pilot and every potential crew could be actively employed and uniquely tracked as a crew. To fully employ all the potential crew members using the system-oriented approach, either spare system organizations are required, or the simulators (or any independent crew training device) must acquire their own system organizations. The practicality of this latter alternative is questionable.

Notice that in building the strike package, no new organizations (with new Org-IDs) were required. This is because the correct set of doctrinal, crew, and billet organizations was selected, based upon the unit's potential deployment options. Therefore, the unit was able to task organize using only its predefined DOO structure. This simplifies the building of the air tasking order (ATO) because the strike package is guaranteed to have a world-wide, unique identifier: the Org-ID of the element chosen to serve as the strike package root node that already exists in any database that contains the fighter squadron's DOO. To build the strike package, existing organizations were re-linked to build the required task organization, and these data were exchanged as part of the ATO.

Having a DOO adds tremendous flexibility to the tasks of managing and tracking operational units. Because each organization serves as a world-wide, unique aggregation point for data, any arbitrary fact about the organization has a predefined index point. This means that location data (for example) about any echelon can be explicitly and consistently maintained. In the strike package example, the location of the crew, which corresponds to the location of the aircraft, is maintained and exchanged in terms of organizational information. If an aircraft has an on-board guidance system, then its location is maintained as an attribute of the crew that operates it. The center of mass of all four aircraft in Strike Package 53 can be maintained under Element B's Org-ID.

Should Package 53 need to split into two groups, as illustrated in Figure 29, then an explicit set of links is modified. First, another (unused) element of the flight is selected and given an alias (e.g., Package 53B). Then, two of the four crews are attached to the new element and a leader is identified and explicitly annotated as in the previous example. Now the average location for each of the two elements can be maintained via the Package 53 and Package 53B organizations, while the individual aircraft locations are still maintained via Crews 1 through 4. Merging the two strike packages together is accomplished by reversing this process. All these actions can occur according to a prescribed set of standing operating procedures (SOPs) based on TTPs.

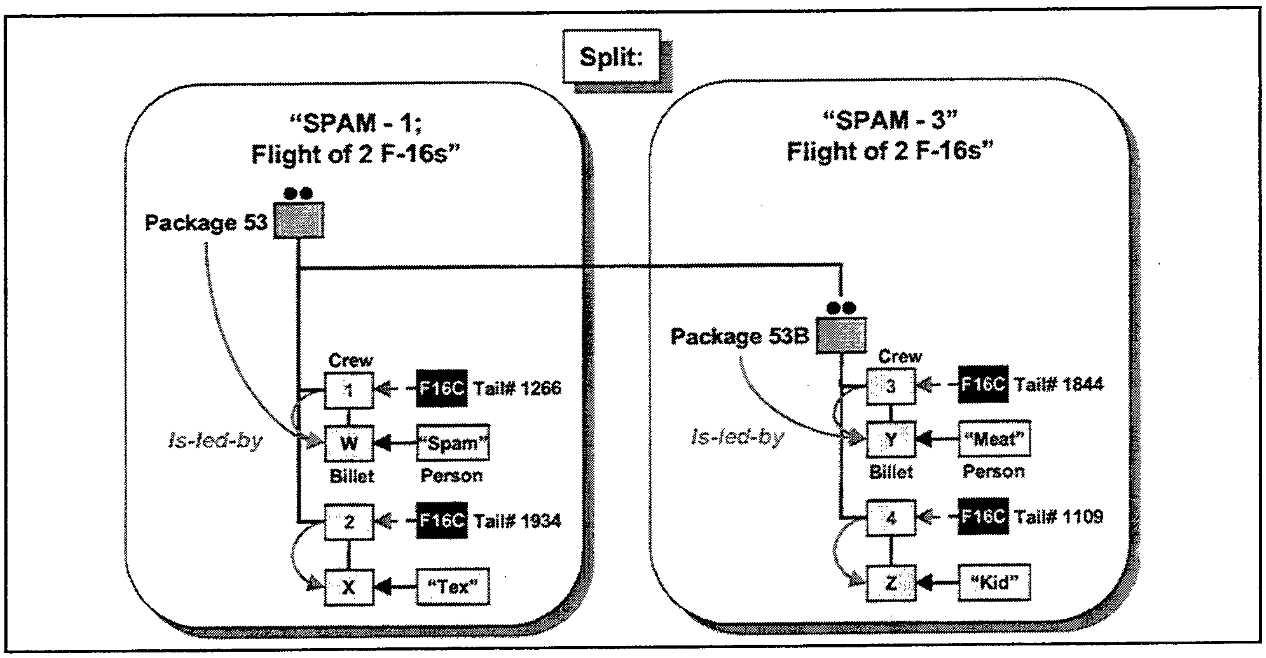


Figure 29: Strike Package 53 Split During Operations

2.4.1.2 Naval Examples — Administrative and Operational Command Structures

The primary design objective for DOOs is to provide a set of data abstractions and processing that are completely homogeneous and general in nature so that they can be applied seamlessly to any task organization problem. It should be clear that the task organization process is equally applicable to large organizations, such as a carrier battle group (CVBG) or an MEU, as it is to small unit operations such as strike packages.

Figure 30 illustrates a simplified view of a Marine Corps force and depicts two categories of units: one set that provides the training and administrative function and one set for deployment. Note that there are separate commanders for each function, which is in contrast to the Army approach wherein one commander does both jobs. The MEU makes judicious use of predefined doctrinal links (see Figure 5) and explicitly identifies membership in multiple command structures (or chains of command). In this example, battalions and squadrons are first task organized into their deployed configuration: a battalion landing team, a composite squadron, and a service support group. This requires several additional OTORs to represent the temporary augmentation of selected units. Then the three task organized units are attached to the MEU via the three predefined roles for the GCE, ACE, and CSSE, respectively. All this reorganization is represented with a few dozen OTORs that are added to the computer org-association tables (see Figure 4). This is a very terse process that requires little bandwidth and could be implemented via the digital operations order.

Bear in mind that although new (temporary) links have been added to represent the attachments, the default assignment links still remain intact. Technically, this means that the organizations are members of two command structures at one time; they have their assigned (or default) parent, and they have an attached (or operational) parent. It is common to be a member of several command structures at one time. The OTOR types determine how an organization responds

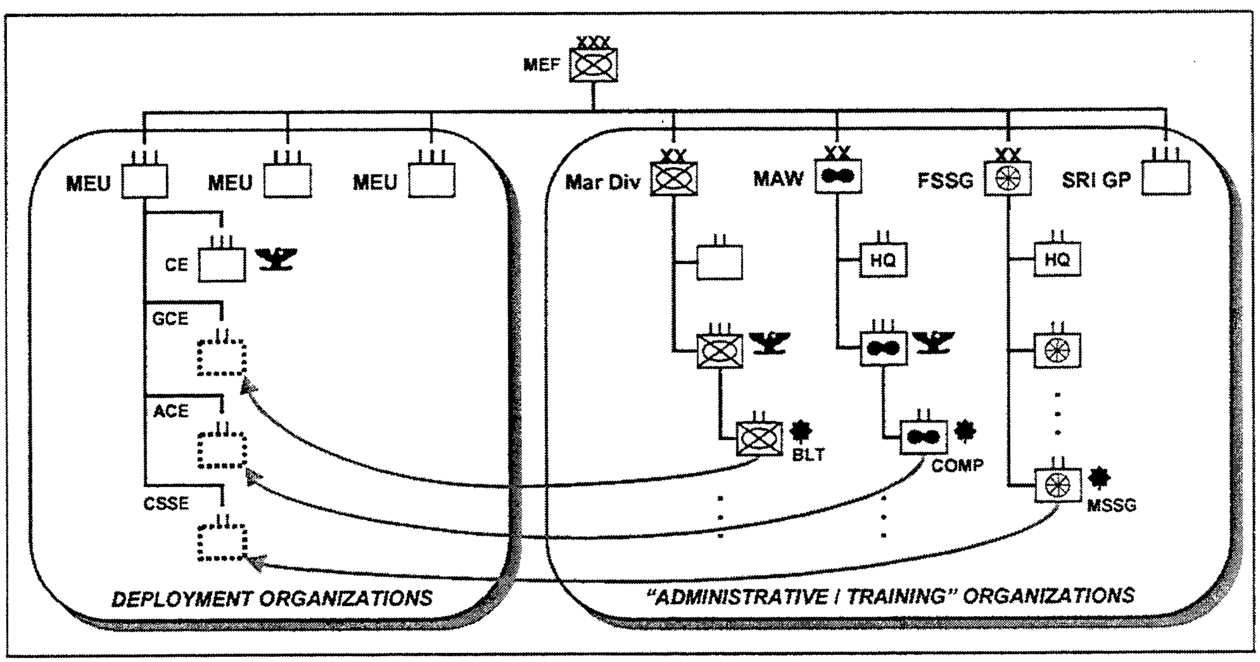


Figure 30: Membership in Multiple Command Structures

to its multiple parent links. For example, the BLT core unit (a infantry battalion with a fixed Org-ID) remains permanently assigned to an infantry regiment; however, in its role as a part of the MEU, the BLT commander regards the MEU commander as his boss (for all practical purposes) for the duration of the deployment.

Figure 31 illustrates the next step in task organization when an MEU is further attached to its amphibious ready group (ARG). In this example, ships (i.e., their crew organizations) are attached via OTORs to the skeleton ARG root organization that is a place holder, just like an MEU, for attaching deployable units. Personnel from an amphibious squadron (e.g., CPR-12) provide the naval component of the command infrastructure for the ARG organization. Whether a command element is permanently in place, as with an MEU command element, or its resources reside under another organization until deployment, as with an ARG, is irrelevant. The same processes apply in either case. As with an MEU, habitual doctrinal links or roles can be predefined to assist in the deployment definition and implementation process (e.g., the roles of "HQ" and "MEU" in Figure 31).

One must not be confused by the fact that traveling on board an asset does not imply that one is a member of the crew of the asset, at least not in the technical sense. Although an MEU may deploy aboard a ship, it is not part of the ship's crew, although it is part of the ARG. Additional interactions, such as *is_embarked_on*, can be added to represent these types of relationships. Another common example of this is an Air Force aircraft full of Army paratroopers. The C-130 aircraft that carries the paratroopers to the drop zone has a crew that is composed of a pilot, copilot, flight engineer, and load master. The soldiers in the back of the plane belong to a different command structure than the crew in the front. However, within the aircraft, there is a command structure based upon the fact that, regardless of rank, pilots are in charge of their aircraft during certain phases of a flight, such as take-off, landing, emergencies, and any other time that flight safety is a concern. Clearly, these rules exist even though they have never been formalized. Exactly how this is to be represented with OTORs will have to be determined by a joint forum. In any case, any formal model of organizational structure must be capable of representing this dual command structure unambiguously if automatic reconfiguration of command systems is to be realized, based upon dynamic changes in a command structure.

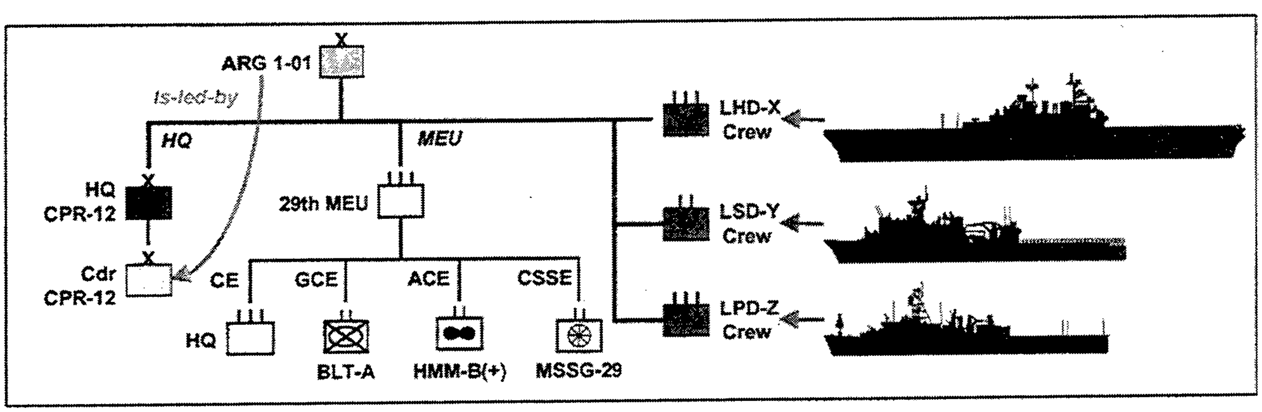


Figure 31: A Simplified View of a Task-Organized Amphibious Force

Another interesting case is illustrated in Figure 32. In this example, the asset (a ballistic missile submarine) has two distinct crews (typically called "blue" and "gold"). For this case, the crew approach provides a more convenient representation than the system approach for several reasons. First, it recognizes the fact that a formal crew structure exists even when the crew is not deployed (i.e., aligned with the asset). Second, it provides a simple facility to shift alignment of the asset between the two crews as they deploy. Finally, it supports the structure of DOOs by allowing habitual doctrinal organizations to be predefined into the structure. Any organization that is routinely required may be added as a doctrinal organization place holder. It does not matter if it is often "empty" (i.e., it does not have any permanently assigned subordinate organizations), as was described with the ARG example. The same is true at the lower echelons. Good examples are command posts and watches that may only be "filled" during deployments but may always have equipment aligned with them. The crews depicted in Figure 32 include the typical administrative structure (highest echelons shown) and a watch. The purpose of the watch sub-structure is to provide an anchor point for the explicit representation of the operational command structure that includes habitual intermediate levels of aggregation (i.e., doctrinal organizations) and ultimately, the attachment points (i.e., doctrinal links or roles). Thus, even within the microcosm of a crew, a DOO and an operation command structure can coincide.

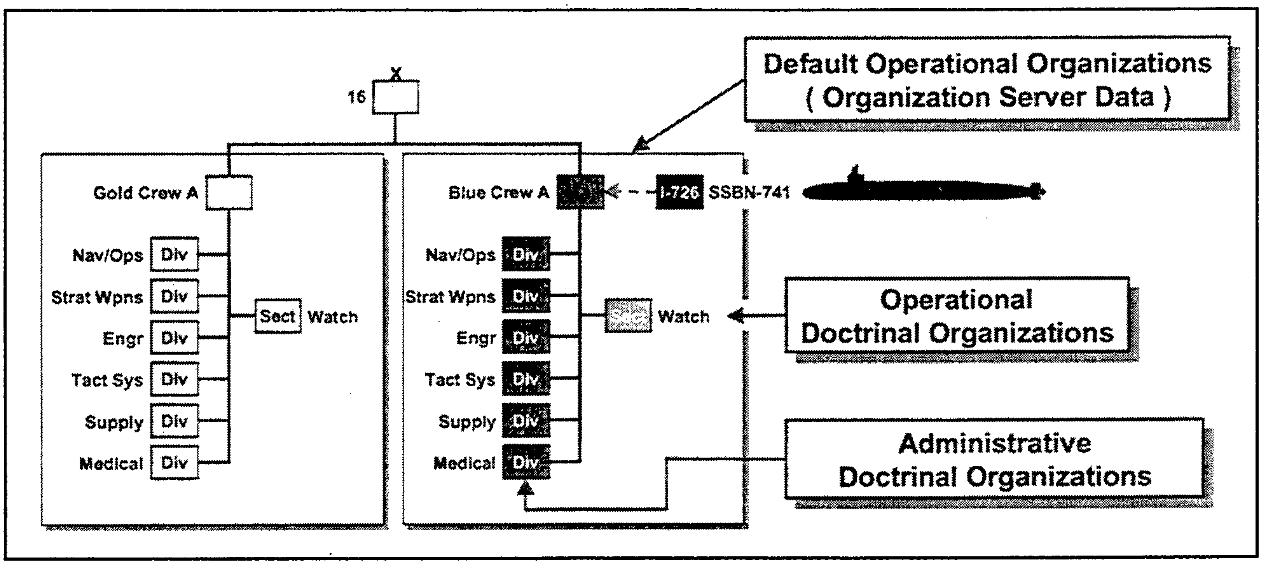


Figure 32: An Asset With Multiple Habitual Crews

A very detailed, well-documented example of simultaneous, multi-command (e.g., dual) structures exists in Navy SMDs. The Navy has traditionally operated with two concurrent, explicit command structures: one administrative and one operational. Figure 33 illustrates the major sub-elements of the administrative command structure of a typical DDG-51 crew as documented in an SMD. The darkly colored boxes are doctrinal organizations, while the lightly colored boxes are officer billets (shown here to emphasize key positions). This command structure includes all the billets for the sailors on board the ship, along with the required qualifications of the billets.

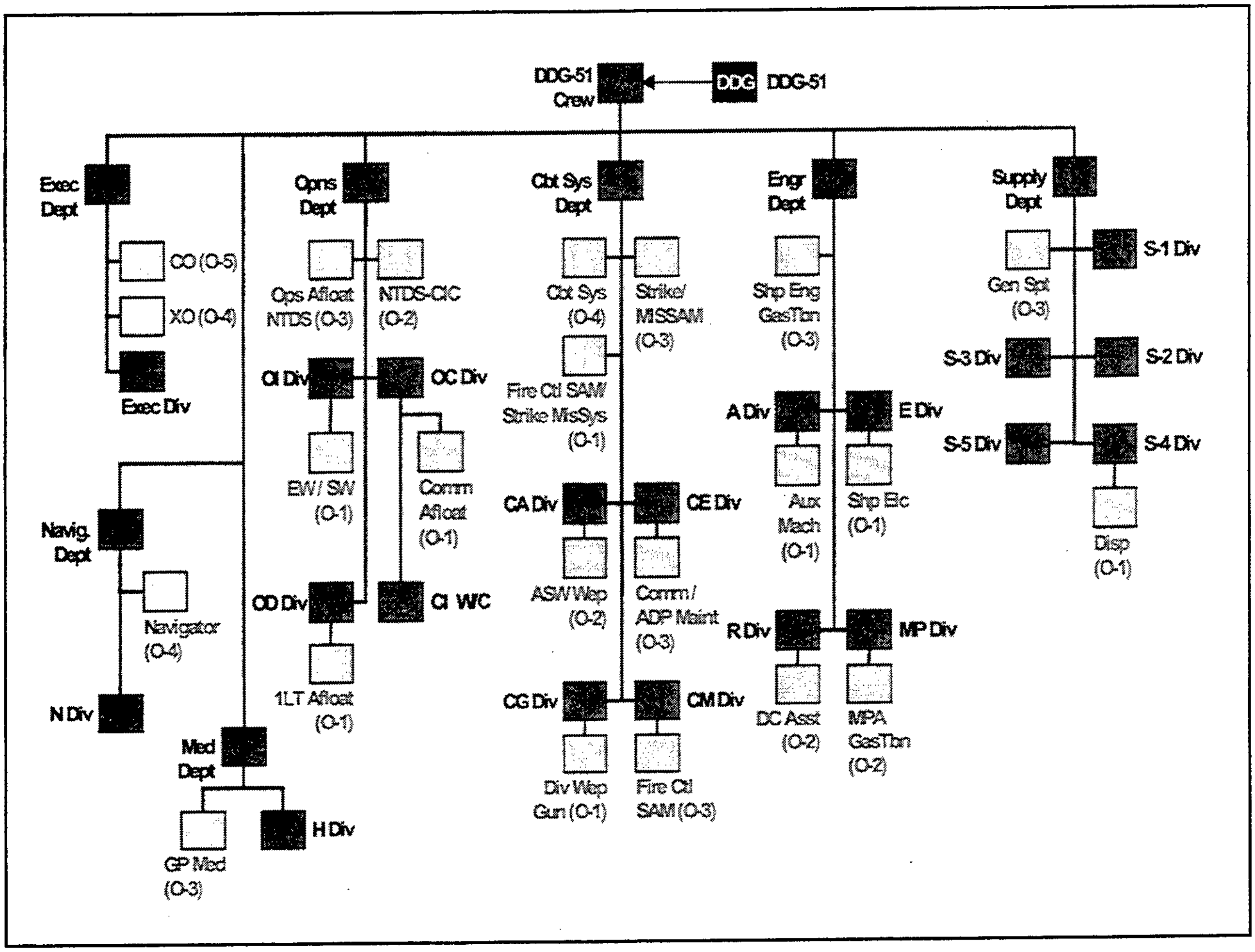


Figure 33: Administrative Structure from an SMD

Figure 34 displays the ship's concurrent operational command structure, called the "battle bill." As with the administrative structure, doctrinal organizations serve as aggregation points for command and functional purposes. The solid boxes represent the primary doctrinal organizations depicted in the SMD. However, there are no billets in the battle bill because having them in both command structures (administrative and operational) would be confounding. Instead, doctrinal links (roles) are used to link the leaves of the operational command structure to billets. These roles, called watch stations, identify the positions that are to be assumed by people in billets (represented by dashed lines). In other words, watch stations are links (roles), not nodes. Like billets, roles can also have a list of qualifications and constraints required to fill the role. Ideally, the qualifications of the person in the billet match the qualifications of the role, but this may not always be the case during personnel shortages.

A watch is an instantiation of this structure with specific billets attached to each role and is time phased throughout the deployment cycle. To tie the two command structures together, the root node of the operational command structure should be labeled "watch" (or some other name) and be made a child of the root node in the administrative structure as in Figure 32. Thus, conceptually, watches are filled by temporarily attaching billets to predefined roles in the operational command structure. This is the identical process used in all the examples in this section.

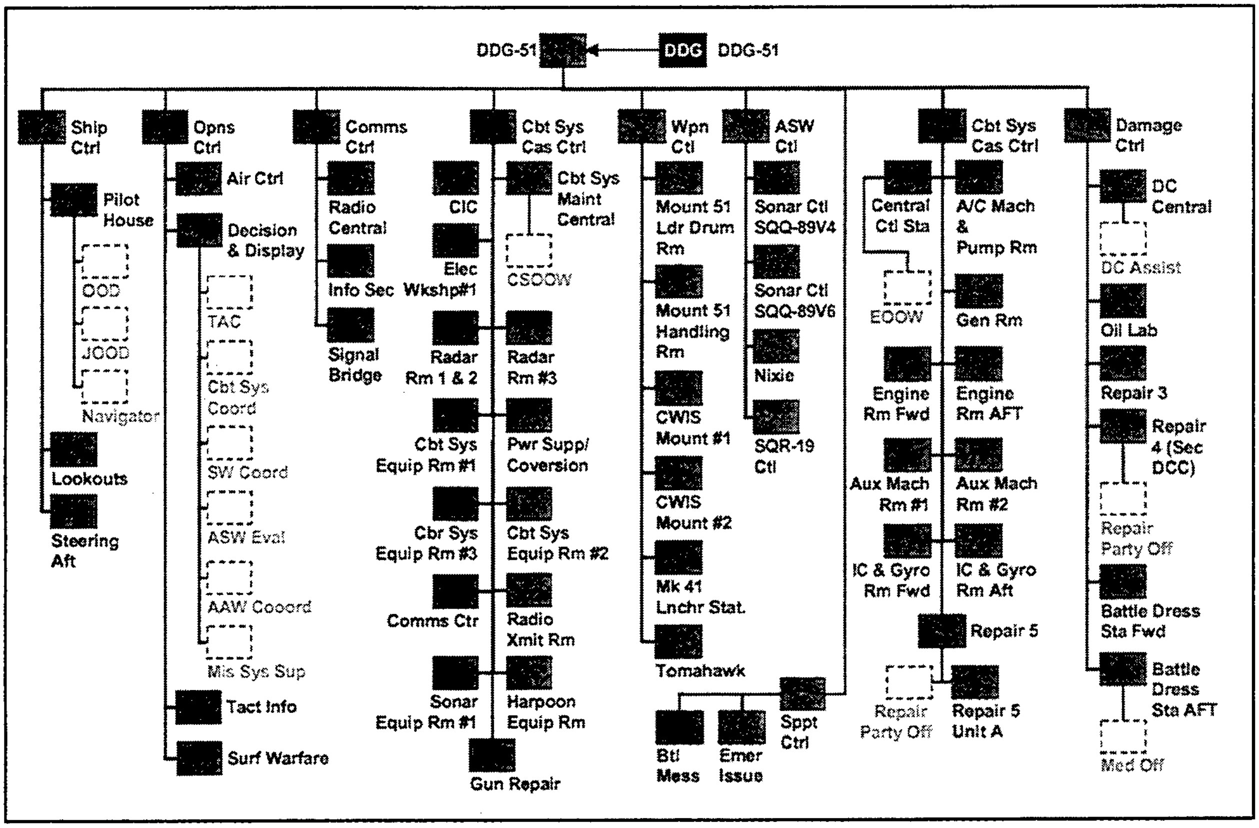


Figure 34: Operational Command Structure - Battle Bill (Watch) Control Stations

2.4.1.3 Army Example — Aviation Packages

Although fixed and rotary winged aviation assets have many similarities, especially in deployment procedures, they are organized quite differently between the services. Figure 35 illustrates the current configuration of an Army attack helicopter company. Several features distinguish it from its counterparts in the other services. First, it exhibits a definite crew-oriented because there is a one-to-one correspondence between flight crews and assets. There are eight aircraft and eight flight crews. This is in contrast to an Air Force fighter squadron that has more pilots than aircraft and a Navy tactical electronic warfare squadron that has more aircraft than flight crews. Second, aircraft and personnel are consolidated under a common organization at much lower echelons (e.g., a platoon versus a squadron) with a junior officer as a leader (the platoon leader). Unlike an Air Force flight, pilots serve as basic maintenance officers, and the operational platoon includes the basic maintenance crews and crew chiefs; however, a separate maintenance company (of the parent battalion) handles more complex maintenance tasks. Finally, it is interesting to note that the aircraft crew members are identified by their secondary specialties. Thus, a person is the aviation life support equipment (ALSE) Officer and also a pilot. Therefore, although Army aviation assets are non-habitually aligned for all the same reasons as other service aviation units, the Army aviation

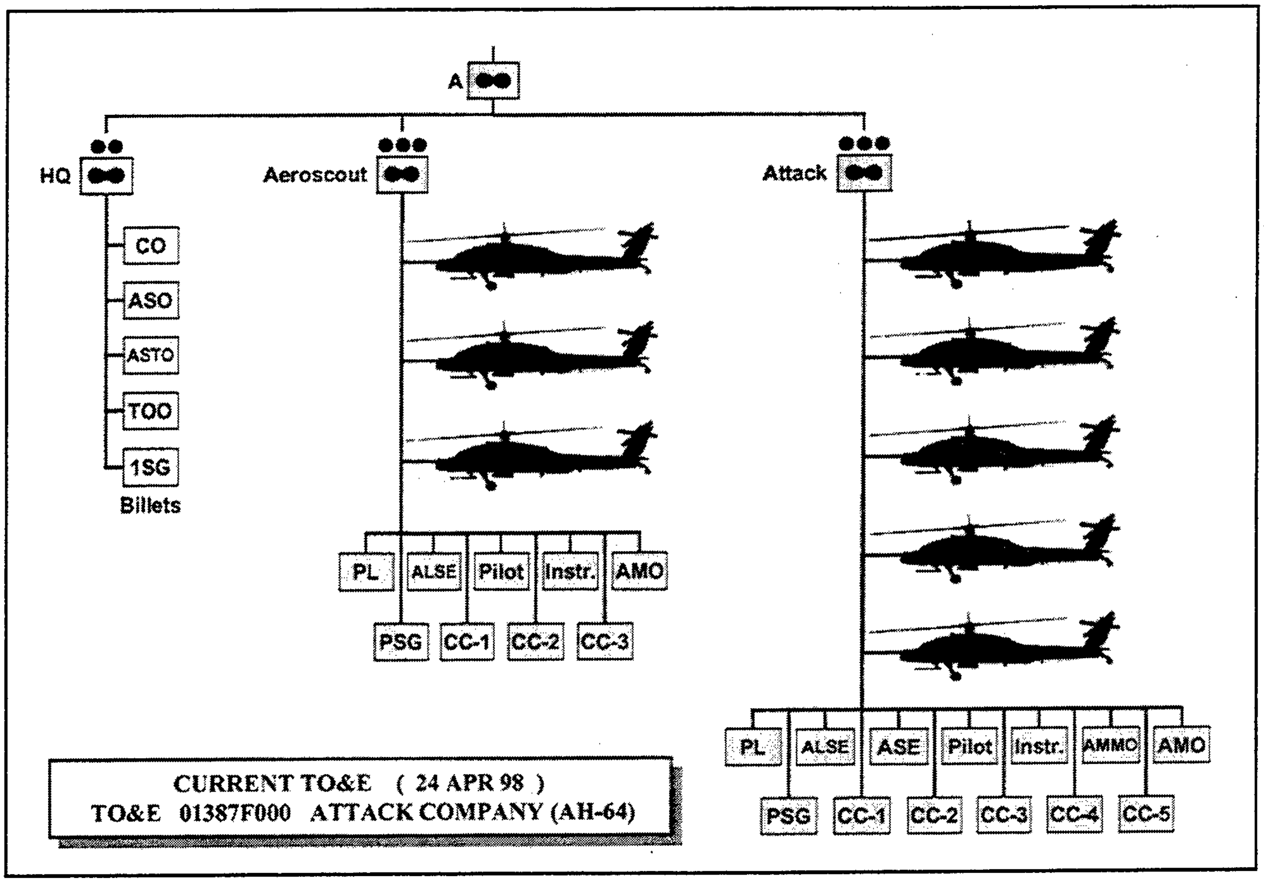


Figure 35: Current Army TO&E Structure of an Aviation Company

organization appears very similar to its ground component siblings in the style of its structural composition.

Like most other staffing documents across the services, Army aviation staffing documents are missing the lower echelon internal nodes (i.e., doctrinal organizations and crews) that explicitly identify operational aggregation points that make the document valuable to deployed or fighting units. Figure 36 illustrates the same organization as Figure 35, but it is augmented with doctrinal organizations and crews; in other words, it is converted to a DOO. Notice that nothing has changed as far as the physical composition of the unit; it has the exact same equipment and billets as before. All that has been changed is the addition of a few new organizations to explicitly reflect the way the unit fights and deploys. This example is one of many possibilities.

First, crews are added, one per each pair of pilots. Since there are eight pairs of pilots, there are eight crews. Notice that from the diagram, there is no way to determine if the number of crews is based upon the number of assets (aircraft) or the number of crews (pilot-gunner teams). It is a coincidence that they match. However, if more aircraft were added, there would still be only eight flight crews. This is opposite of the system-oriented approach, in which more system organizations would be added if more aircraft were added, even if the number of flight crews remained the same.

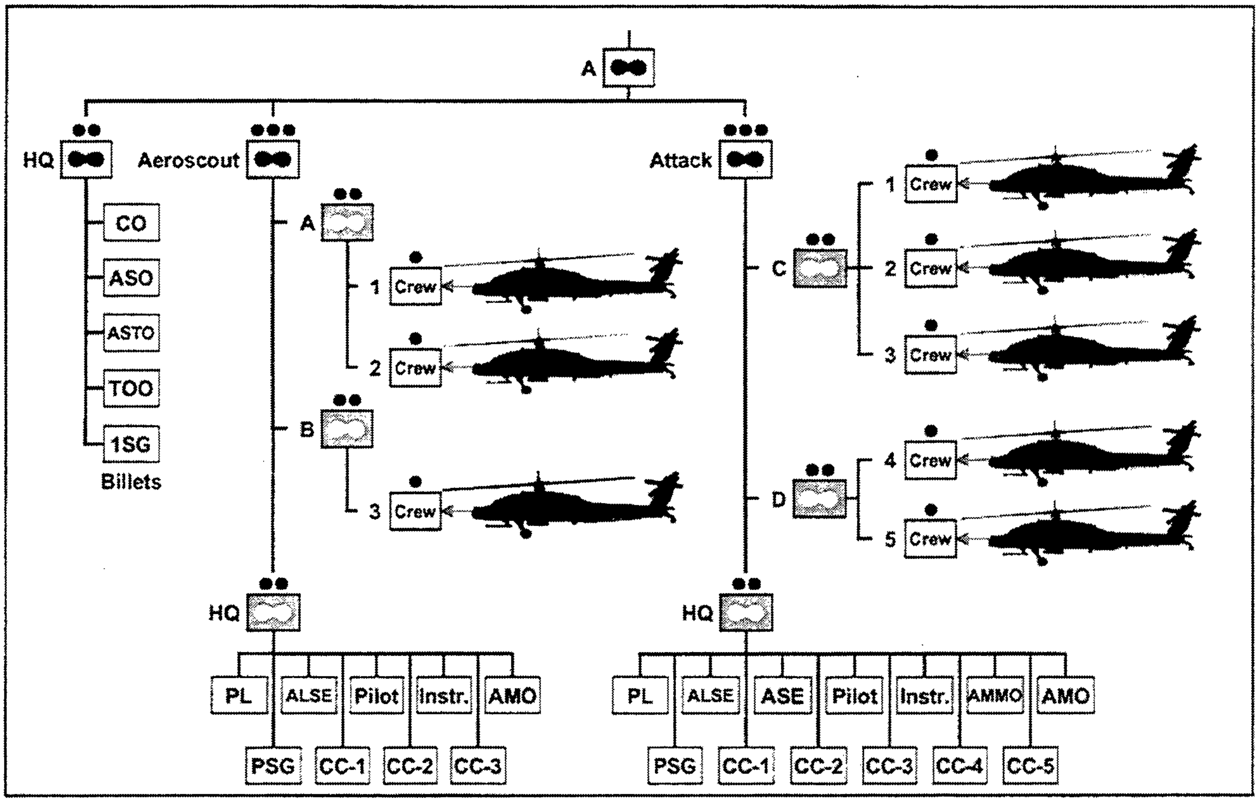


Figure 36: A Notional DOO of an Army Aviation Company

Second, doctrinal organizations are added at common aggregation points. In this case, adding one doctrinal organization for every two aircraft explicitly recognizes the heuristic of fighting in pairs (i.e., "lead" and "wingman" teams). For this example, these "sections" have been identified with the simple titles "A" through "D." As was mentioned earlier, this binary association guarantees that any combination of aircraft can be aggregated and tracked. If all eight aircraft are individually deployed, the crew organizations serve as the aggregation points. For pairs, the aggregation points are the sections. In fact, any number of aircraft can be attached to a section and tracked as a group.

Clearly, this is just one of many ways to build the DOO. For example, the crew organizations could be associated with the maintenance crew chiefs and placed under the "HQ Sections." There could be separate crew organizations for both maintenance and flight operations, as was illustrated in Figure 23. There could be a maintenance platoon and an operations platoon, perhaps each with two sections. The options are endless for force development experts to explore. All are viable, provided that they meet the requirements of the operational user for aggregating (and therefore tracking) arbitrary groups of organizations. However, bear in mind that the same problem identified with other non-habitual assets also exists in this case: how to represent the non-habitual assets when they are not being used. The option of having separate crews for maintenance and operations handles this situation well.

Figure 37 illustrates using a DOO as the starting point for building an OOB. Notice that the term OOB is used in its most general form to mean any modification of the DOO. In this example, it includes both the attachment of a crew to a different section (i.e., temporary attachment of "Crew 3/C" to Section B) and assigning people (in billets) to crews (i.e., assigning the pilots in the billets of

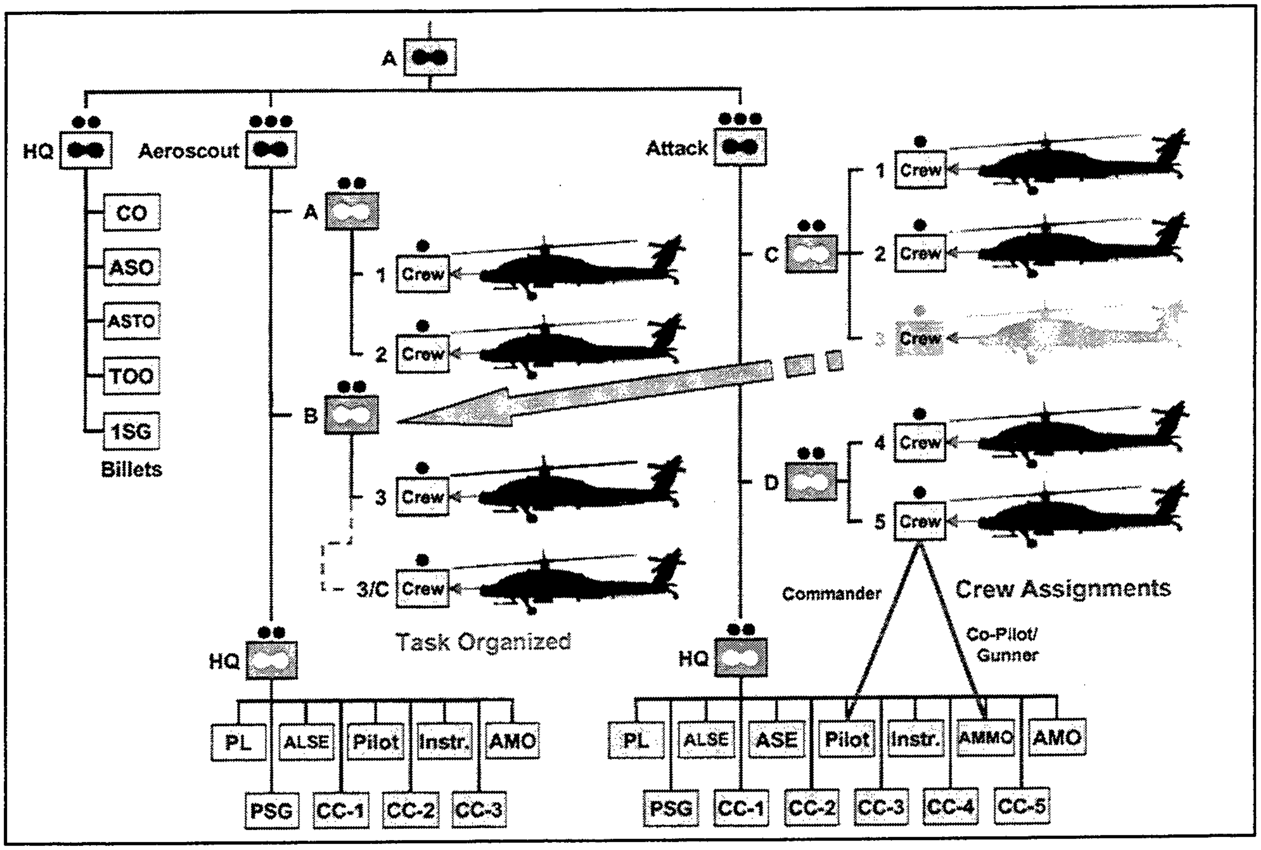


Figure 37: Using a DOO to Build an Order of Battle

Pilot and AMMO under the Attack Platoon to the roles of "Commander" and "Co-pilot-gunner" of Crew 5). The advantage of a general approach is that the same set of algorithms can be used for many tasks, such as scheduling flight time for training, task organizing for combat, or assigning staff duty or watch officers. Since the organizational tree structure is homogeneous down to the billet level, any arbitrary attachment can be established, provided that the required aggregation points exist. This is the task of the force development community, who must work with the military user to astutely consider where to place doctrinal organizations.

3 UNIVERSAL NAMING

Recall that a *surrogate key* is a primary key for a database entity that has no inherent meaning, is composed of only a single (database) field, and whose value is not derived from any other entity (i.e., is non-identifying). Integers make good primary keys because they are simple, easy to manipulate by machines, and easy to verify for uniqueness. An Org-ID is a surrogate key that uniquely identifies a node in an organizational tree.

Ultimately, one wants to uniquely identify all the entities distributed across the battle space via myriad computers. Imagine that the primary keys of all entities are surrogate keys, that is, they are integers. This is in stark contrast to the usual method of defining primary keys by building them from numerous other primary keys (as in IDEF1X data modeling). Using surrogate keys, the size of the keys could be a constant, based on a "large integer" (e.g., in the 64- to 128-bit range).

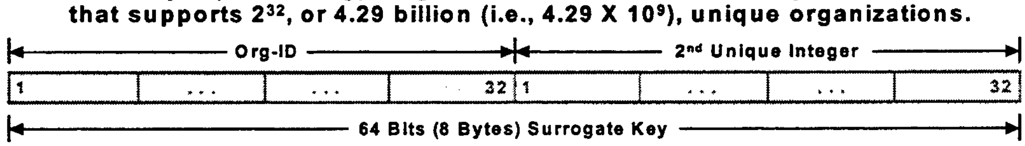
The challenge is to guarantee that no two integers are reused, at least within a given time frame. This is where Org-IDs help. Conceptually, every organization is allowed to create new entities and distribute them to other databases. Org-IDs are guaranteed to be unique, so if a database obtains its "identity" from the organization of which it is a part, then one can be assured that the database has a universally unique identifier. Universally unique surrogate keys have been recently described in Johnston (2000) where they are called "enterprise keys."

Unique surrogate keys can be composed by combining multiple unique identifiers. If a unique number is appended to an Org-ID, the resulting number is also guaranteed to be universally unique. Therefore, a database can create a universally unique surrogate key by simply creating its own unique number and concatenating it with its Org-ID. In other words, when a new entity is created, the database on which it is created assigns a surrogate key to the entity that is the concatenation of the Org-ID of the organization that controls the database and another integer. The database management system's only responsibility is to ensure that it never uses the integer it creates twice, at least within a specific time frame. This is illustrated in Figure 38.

A surrogate key is a machine-generated, primary (database) key that is forever bound to a data item at the time of its creation. Therefore, an entity's surrogate key (or object ID) is an integral part of its structure and moves with it as it traverses numerous computers. In other words, the surrogate key that is assigned when an entity or object is created stays with it as it propagates among battle command systems. If all systems use Org-IDs for their identity, then a surrogate key composed from an Org-ID is guaranteed to be unique, regardless of the database to which it is disseminated. This will be true for any organization using Org-IDs, including coalition partners, non-government organizations, and private voluntary organizations.

Key size is always an issue for debate. One characteristic to keep in mind is that identifiers are not addresses. They simply provide a set of bits to uniquely identify something. Therefore, there is no need to distribute them in blocks (like telephone area codes or Internet Protocol [IP] addresses),

A world-wide, unique surrogate (primary) key to identify *any item* in *any database* can be composed by combining identifiers (meaningless numbers).
First, a 4-byte (32-bit-long) "organizational identifier," or Org-ID, is provided that supports 2^{32}, or 4.29 billion (i.e., 4.29 X 10^9), unique organizations.



If a database obtains its identity from the organization to which it is a part, then a second (4-byte) number that is *controlled by the database* (and provides another 4.29 billion unique entities for that organization's database users) can be combined with the Org-ID to provide a new, bigger unique number.

Thus, some day, there could be a common primary (surrogate) key structure of 8 bytes (64 bits) that would allow 2^{64} bit patterns = 18.45 X 10^{18}, or 18.45 Exa-keys, or 18 billion billion unique entities to be tracked.

Figure 38: Surrogate Keys Composed of Composite Integers

thus vastly reducing the potential for waste. One should expect every possible identifier to be used before any are recycled; for a 32-bit identifier, this is nearly 4.3 billion combinations. However, to alleviate any anxiety, the top bit of an identifier can be reserved to indicate future expansion. This results in a 50% reduction of combinations. Thus, a 32-bit identifier is reduced to providing 2.1 billion unique identifier but with the additional capability to expand indefinitely and provide tremendous flexibility to the implementers (e.g., the first bit indicates if another 4-byte integer follows, etc., to allow unlimited chaining).

Many of the data in a battle command system are reference data (e.g., the range of an F-15 with wing tanks, the existing ships within the Navy, or on a larger scale, the existing organizations within DoD). In general, reference data are relatively static and consequently, many of them can be predefined and, as required, pre-loaded into computer databases with pre-assigned USKs. Ultimately, it is expected that servers containing a wide variety of structured, as well as the currently unstructured, information will be readily available to battle staffs before and during military operations. This is illustrated in Figure 39.

Several major efforts have been commissioned to formally describe and document schema for battlefield objects.[30] These include (but are not limited to) the DoD Command and Control Core Data Model (C2 Core DM), the ATCCIS Generic Hub, and the DoD C4ISR Core Architecture Data

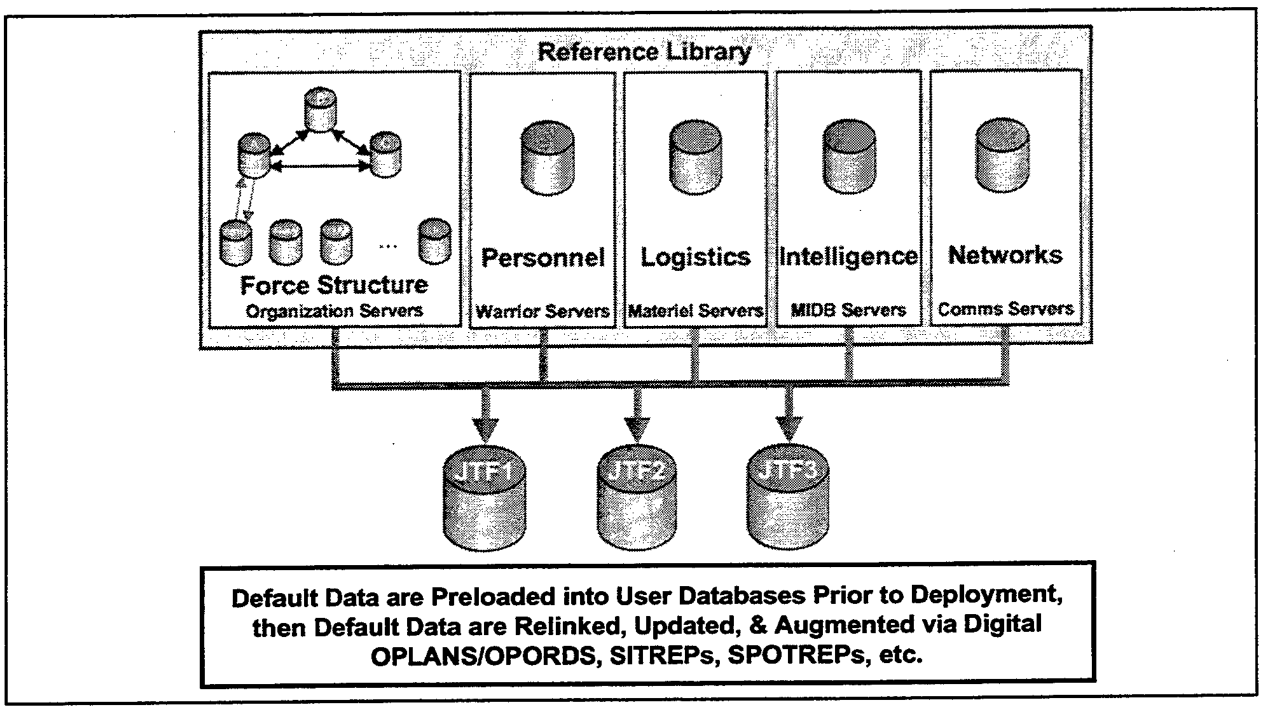


Figure 39: Ultimately, Default Data Available for Download from Many Sources

[30] Schema: a representation of the structure of data.

Model (CADM).[31] All these products have detailed descriptions for the structure of the data; however, a separate, major challenge yet to be adequately addressed is the population and maintenance of the data themselves.

A valuable feature of the USK approach is the ability to facilitate concurrent population of databases from many distributed sources. For example, a virtual materiel server might contain equipment data for all the services. In actuality, many servers would be partitioned by service and function. In the materiel community (as in most others), data creation falls basically along organizational lines. One organization may be the definitive source of information about Air Force jet aircraft, another about Marine Corps amphibious vehicles, another about Army rotary wing aircraft, and so on. Each of these domains may be partitioned even further, resulting in many existing (legacy) systems serving as the source of data for each of these sub-domains. To support the population of the joint common database (JCDB), one of the products of each of these disparate systems could be database records in a common schema that collectively comprise the virtual materiel server.[32] By using a USK for a *materiel identifier* (mat-id), each separate organization can enter data without fear of having its primary keys collide with anyone else's.[33] Therefore, database population and maintenance can be distributed and accomplished concurrently among all the data providers.

Each legacy system can retain its own key management system, but the mat-id is used to integrate across service and function boundaries. In other words, the mat-id can be an alternate key for the records stored in the various databases. When outside organizations require data, the mat-id is used to distinguish and associate data entities. For example, when force developers require materiel data to describe the outfitting requirements of their DOOs, they go to the materiel server to obtain the unique mat-id for the equipment they align (via ETORs) with their organizations. The same is true for other domains.

This approach is equally applicable for dynamic data as it is for reference materiel. When a reconnaissance team enters a sighting into their wearable computer, the data are tagged with a USK. No matter where those data propagate, they are guaranteed to have a unique identity. This does not solve the fusion problem; that is, it does not prevent two teams from seeing the same item and reporting it twice, probably with different details. It does ensure that the two data items (reports) are uniquely identified and provides a simple, terse, bandwidth efficient way to audit the data fusion and collation process in a manner conducive to machine manipulation.

If two items of data have the same USK, they are semantic equivalents. Thus, USKs can serve as the common naming convention between disparate, distributed information systems, regardless of whether text data are in French, English, German, or any other language, or whether the database is in a relational, object-oriented, or other form.

---

[31] C2 Core DM: http://www-datadmn.itsi.disa.mil/ddm.html
Army Tactical Command & Control Information System (ATCCIS) Generic Hub 4: ADatP-5 Draft 1.0, *The Land C2 Information Exchange Data Model*, 1 Oct 1999, http://www.euronet.nl/users/atccis/index.html
CADM: http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/index.htm

[32] A database that uses the C2 Core Data Model schema and serves as the centerpiece for Army Force XXI data storage.

[33] Recall that a USK is composed of an Org-ID concatenated with another unique integer of the creator's choice.

A word of caution: Although the purpose of the surrogate key is only to provide a unique identifier (i.e., one that is guaranteed to be universally unique), this approach has a secondary benefit: one can identify the organization that created the entity or at least, the organization that claims to be the creator.[34] Whether this feature should be exploited is questionable because using this information violates the basic tenet that a surrogate key should be meaningless. Abusing this policy can easily lead to problems because unanticipated conditions may arise in the future, which invalidate the assumptions about the meaning presumed for parts of the surrogate key.

3.1 Special Cases: Enemy and Ad Hoc Organizations

Two issues that are often raised are (1) whether enemy organizations receive Org-IDs and (2) what happens if soldiers immediately need unanticipated organizations. The discussion of these issues has been postponed until this section because the answer relates to USKs.

3.1.1 Enemy Organizations

Enemy organizations do not get Org-IDs. The reason is that enemy organizations do not participate in the Org-ID process. They do not build organization servers and register to get Org-IDs from the Org-ID server. Instead, someone else is entering *perceived* data about enemy organizations. Therefore, although enemy organizations may have many of the same attributes and associations as (friendly) organizations, they reside in separate tables.

An enemy organization receives a USK. In this case, the first part of the identifier is the Org-ID of the organization (e.g., billet) authorized to create the data and the second part is a value of the database's choosing. There may be several perceptions of an enemy organization, and this approach explicitly recognizes this fact and provides unique identifiers for each different supposition or estimate. The surrogate keys provide a simple and convenient facility for linking the different perceptions together. As with any fusion problem, an inherent capability is supported to track and audit the combination of different perceptions from several sources as new and more complete estimates are developed.

Two other reasons for having a separate enemy organization table are (1) since it is a perception, there will be different and/or additional attributes in the entity definition, and (2) because the source of the data may be sensitive, the table may have a higher security classification and reside in a separate security domain (such as the intelligence server in Figure 39).

3.1.2 Ad Hoc Organizations

The issue of creating units "on the fly" has already been indirectly addressed by the topic of doctrinal organizations (see Figure 26). Optimistically, a DOO contains all the doctrinal organizations required to handle aggregation requirements for deploying units. However, this will only occur after several iterations of real-world deployments that produce feedback to the force development community about DOO deficiencies, misinterpretations, or oversights. Presumably,

---

[34] As with many other systems, surrogate keys can be easily modified by proxy agents to hide the original identity of the information's creator.

there will always be unpredicted situations or configurations, so there must be a procedure to allow one to quickly create a new organization to serve as a temporary doctrinal organization (i.e., aggregation point). There are three basic alternatives: on-the-spot requests for new organizations, pre-allocation of spare organizations, and ad hoc organizations.

On-the-spot requests are not practical. First, the communications delay incurred at low echelons to reach the service's organization server would be prohibitive. Second, one cannot expect a non-force development person to know the details expected to correctly add organizations to the force structure, and third, since these are typically temporary organizations, it is wasteful to add and then delete the organization from the official organization server. Consequently, this avenue is discarded.

Pre-allocation of spare organizations is a viable alternative. In this case, the spare organizations are part of the official DOO, and by using aliases, one can attach any name to a spare organization for the duration of its use. The number of spare organizations required to guarantee that any level of aggregation can occur can be calculated. Recall the previous discussion of binary trees in which it was explained that this configuration guarantees that any desired aggregation can be accomplished. Also recall that half the nodes of a binary tree are in the leaves and half (minus one) are internal nodes (i.e., crew or doctrinal organizations). Therefore, to determine the maximum number of spare organizations required for any given echelon, one simply subtracts the number of doctrinal and crew organizations specified for the unit from the number of billets. Therefore, if an organization has 200 billets, then 100 internal nodes are required to provide any possible configuration. If there are already 70 crew and doctrinal organizations (combined) in the DOO, then a maximum of 30 spares is required to handle any possible situation (i.e., $200 \div 2 = 100$; $100 - 70 = 30$).

The main issue is the balance between control and responsiveness. This is really a question of the echelons to which spares are pre-allocated. At one extreme, they can be allocated to the lowest echelons available, for example, at the Army squad level as illustrated in Figure 26-C. This provides the ultimate responsiveness but also provides the least control. It is easy for untrained personnel to incorrectly use (or misuse) spare organizations. At the other extreme, spares would be maintained at the Army division level. Now there is excellent control, but responsiveness would be poor. Obviously, the answer lies somewhere between two extremes and can be determined and adjusted with experience. A good compromise would be to maintain spares at two echelons; for example, one set could be maintained at the Army division level to handle company-sized organizations and larger, and other sets could be maintained at the company echelons to handle requirements for the echelons below them. The normal procedure would be to base the allocation of spares on expected mission requirements (e.g., number of listening posts assigned to a platoon) and to distribute them via the operation plans. Even in unexpected situations, however, no more than three echelons would have to be traversed to obtain additional spares.

The third alternative is *ad hoc organizations*. An ad hoc organization is one that is not included in a DOO; in other words, it is not maintained in an organization server. Consequently, ad hoc organizations do not receive Org-IDs. Instead, they are assigned USKs, as are enemy organizations. As with enemy organizations, ad hoc organizations are maintained in a separate table and may have some different attributes than organizations. As expected, the first part of the USK is the Org-ID of the organization authorized to create the ad hoc organization, while the second part is any number chosen by the database upon which it is created. Since ad hoc organizations are not reference materiel, it is the creating organization's responsibility to disseminate the information to all required recipients before any ad hoc organizations are used.

The decision to use doctrinal or ad hoc organizations will ultimately depend upon the situation. Ad hoc organizations provide tremendous flexibility and, as is often the case, the potential for misuse. Therefore, it is the responsibility of commanders and their staffs to judiciously control their creation. Ad hoc organizations should only be used when the traditional approach truly cannot handle a situation. If this identifies a recurring condition, then feedback should be given to the force development community so that the DOO can be modified.

One potential use of the ad hoc organization is aviation strike packages. Although Section 2.4.1.1 illustrated how one can build strike packages using doctrinal organizations, this can also be accomplished with ad hoc organizations. Since the dissemination of the ATO is already a major undertaking, the extra effort to send ad hoc organization information is most likely negligible. However, if one is concerned about interoperability and dissemination with neighboring services, ground forces, or coalition partners, then the additional distribution requirements may be substantial (e.g., if one is required to pass data between a close air support [CAS] aircraft and maneuvering ground forces).

One final issue is the storage size of Org-IDs. Under this proposed naming scheme, all primary keys migrate to USKs, and all USKs are 64-bit (composite) integers (i.e., except for one entity: organizations that have 32-bit Org-Ids). There is something in-elegant about having all keys except one be a common size. Therefore, one might consider allocating 64 bits for Org-IDs even though only half of the bits are used. This would allow application builders to use one data structure for any primary key and would allow interchange of keys across entities to support more flexible applications, especially in the object-oriented community. This does not mean that all 64 bits of an Org-ID would have to be transmitted—only that once inside a database, they are stored that way. A simple protocol could be built to strip unnecessary bits from Org-IDs for transmission.

For example, suppose there is a general entity called "sensing" that is used to store information about observations between entities. This could be electromagnetic (e.g., visual, radar, infrared, etc.), acoustic, or seismic activity. In essence, this entity would associate several other entities via their USKs. One of the attributes of this associative entity would be a link to the object being sensed, called the *target* field. Consider the case when a multi-spectral sensor first detects a seismic signal and determines that the object is some type of ground vehicle. Initially, a USK for a "ground vehicle" object (a materiel-item under the C2 Core Data Model) would be assigned to the target field. As the sensors gain more data, the information evolves with more accuracy and detail, which may result in the object description moving across entity boundaries. The description may change from ground vehicle to a tracked vehicle, then to a particular model of vehicle (both *materiel-items*), to a bumper number (a specific piece of materiel with a serial number), to part of a generic organization (an *organization-type*), to a specific organization. In this example, the USK references four different types of entities. However, this is a simple task because all the keys have the same form, as is illustrated in Figure 40.

Notice that when USKs are used, all primary keys in a database (and across databases) are unique from each other. Any implementation would surely include a mapping of USKs to entity type and perhaps even indices into the tables themselves. Therefore, any retrieval procedure would have to be able to find the entity type, given a USK, which is reminiscent of network databases, shunned many years ago in favor of relational databases. However, such hybrid approaches do have their advantages, especially in the performance arena, but this is a topic outside the scope of this report.

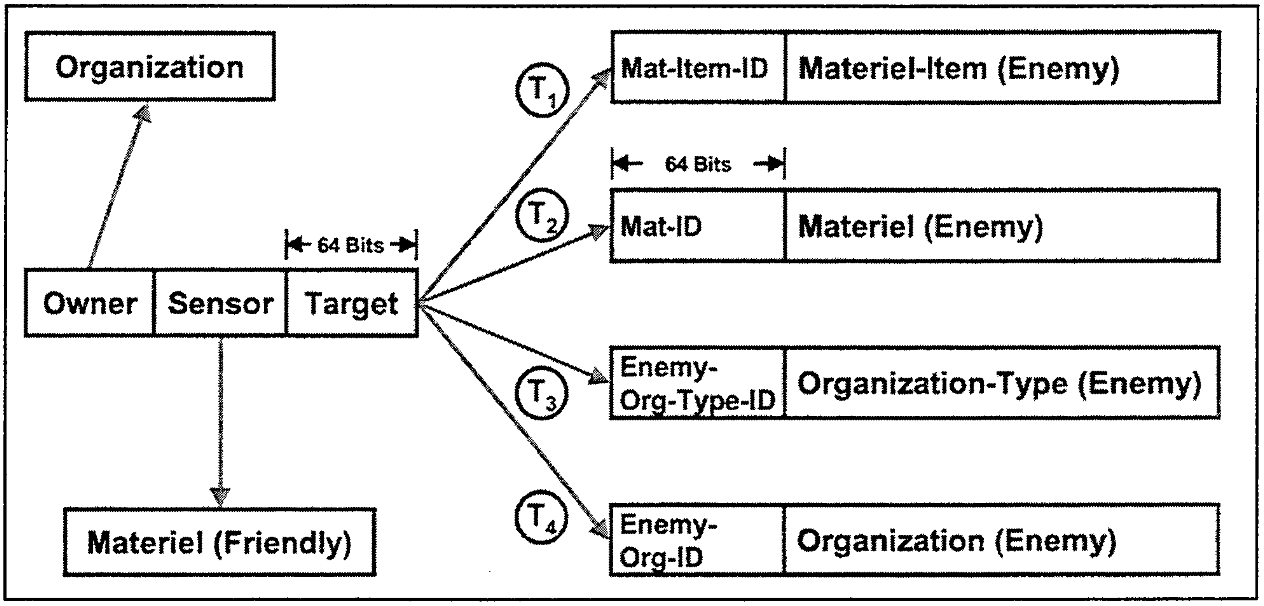


Figure 40: Universal Surrogate Keys Are Interchangeable

4 CONCLUSION

This report presents several major themes. First, it is postulated that the formal concept of organization forms the central framework to which all other battle command entities and functions relate. This means that the organizational structure that defines a military unit provides a framework to which all other battlefield entities can be associated, thus making the organizational data structures the rallying point for the integration of other data, such as those related to logistics, personnel, communications, and planning.

Second, organizational structure is described with tree graph terminology. The following definitions were introduced: the nodes of a tree graph correspond to "organizations"; a set of links connecting a parent node with its descendants corresponds to a "command structure," in which each individual link is a "command relationship"; the combination of the two (a tree graph) is called a "unit." Organization charts are examples of tree graphs. By themselves, however, these definitions are not sufficient. A set of semantics (i.e., meaning of the elements) must also be developed.

To begin the process of developing semantics, a set of hypotheses was presented which states that (1) a relatively stable, default organizational structure exists (2) from which any operational command structure (task organizations) can be generated. In other words, if the force development community designs, builds, and documents the force structure, based on the way an organization deploys or fights, then a default tree graph of nodes and links (i.e., organizations and a default command structure) will exist from which any arbitrary fighting unit can be built by simply re-linking the existing nodes. This means that it should be a rare event to have to create a new node (an organization) for the purpose of attaching other nodes to form a special unit.

Third, a process was introduced to assign and track the nodes and links of these organizational structures. When an organization is officially established by a force developer, it is assigned a permanent *organization identifier*, or "Org-ID," that stays with it for its life (i.e., until it is disestablished). At its creation, each organization is also given a default command relationship to some parent node, thereby adding it to the default command structure. A database, called an *organization server*, is maintained by the force development community, which contains the default structure for the service. These data are downloaded to the operational users as the definitive source of the services' unit data. As the force structure is modified and those modifications are implemented by the operational users, the organization server is updated to reflect the current default state of the service. Therefore, a set of organization servers, maintained by sanctioned force developers, serves as the authoritative source of all unit data for the DoD. How many of the data are shared and with whom they are shared is controlled by the organization server's owner or sponsor.

Separate from the organization servers is a set of Org-ID servers. This is a set of replicated databases that merely dispense guaranteed unique Org-IDs to authorized and registered organization servers. The only data stored in the Org-ID servers are (1) a list of Org-IDs that have been distributed, (2) a reference to whom they were distributed, and (3) whether they are currently in use (i.e., active). The separation of the Org-ID distribution process (via the Org-ID servers) from the Org-ID allocation process (via the organization servers) is essential because it allows participation in the process without relinquishing privacy. Anyone can participate, but sharing how the Org-IDs were assigned is solely the choice of the organization server owners. Therefore, coalition partners can freely participate without sacrificing sovereignty over their unit data, and any organization is free to suppress any unit data that it chooses.

Fourth, a proposed semantics for default operational structures was presented. The concept of *Default Operational Organizations* (DOOs) was introduced as a set of guidelines and rationale for building default organizational structures using general techniques that are applicable across the services. DOOs are tree structures that provide enough detail to meet the requirements necessary to build arbitrary orders of battle (OOB) or military capability packages (MCPs) across service and coalition boundaries. The required detail is provided by extending the concept of organization down to the individual or billet level. By closely studying how each military service organizes for combat, the author developed basic tenets in an attempt to reduce many apparent (but not actual) disparate practices to a few fundamental concepts. The result is a set of recommended guidelines that are based upon the "best practices" of all the services.

Three types of organizations were explicitly defined and described: (1) *billets*, which correspond one to one with people, (2) *crews*, which correspond one to one with major assets (i.e., materiel), and (3*) doctrinal organizations*, which represent routine aggregation points for functional or command purposes (some call this doctrine). From a tree graph perspective, billets are the leaves (or the bottom nodes) of an organization chart; that is, they have no further descendants. A special category of billet, the leadership billet, is explicitly described along with a discussion of the ambiguities encountered when the current practice of "burying them" in the organizational structure is used.

Crew and doctrinal organizations are internal nodes (i.e., non-billets that have organizations below them in a chart). Crews are easy to identify because a major asset (e.g., a vehicle, aircraft, or ship) is involved. The associations between crew members and between crews and their assets are described in terms of habitual and non-habitual relationships. Although these differences have traditionally resulted in disparate perceptions between aviation and surface organizational strategies, they can be coalesced into two basic designs: system (or asset) oriented and crew (or personnel) oriented. This reflects the old adage of "equipping the man" versus "manning the equipment." It is not surprising that the two are ultimately equivalent, and the approach selected is usually based upon the convenience of the application that manipulates the data.

Doctrinal organizations provide the many aggregation points to assemble DOOs and therefore are the key component to building arbitrary OOBs. In other words, without the correct set of doctrinal organizations, the hypothesis that OOBs can be built by re-linking a DOO fails. This is why a DOO must reflect the way one deploys or fights. Currently, doctrinal organizations are manifestations of the "art of force development"; consequently, their selection is often subjectively based upon the personal preferences of the developer. It is interesting to note that most crew and doctrinal organizations that reside in the bottom echelons, just a few levels above billets, are missing from staffing documents. Consequently, force developers must be sagacious in the selection of aggregation points. No matter what the echelon, if it is common to task organize a set of organizations (including billets) into a group, then a doctrinal organization should exist in the DOO to allow it.

A formal difference between a command structure and chain of command is defined. A command structure is a composite structure that connects billets, crews, and doctrinal organization. One interprets the basic meaning of a link as "is composed of." A chain of command only connects billets, and the links are interpreted as "reports to." A command structure can be easily converted into a chain of command if additional *is_led_by* links are established between internal nodes (doctrinal organizations or crews) and a billet. In other words, every active, internal node of an organization chart should have a leader, and a person can be the leader of more than one organization. However, there is not a unique command structure for a given chain of command. It is expected that a frequent requirement will be to toggle between command structure view and chain of command organization charts because one mode of display often identifies and emphasizes inconsistencies better than another.

The DOO is a relatively stable structure that typically changes only over periods of months (i.e., when force structures are updated). Consequently, the data can be distributed ahead of time, as needed, to field commanders. From the DOO, any arbitrary task organizations can be built, down to the individual billet level, by temporarily re-linking existing organizations; often, the new unit is given a temporary alias. The new links are disseminated via the operations planning process. In other words, an OOB is defined by a DOO plus an operations order. Examples are numerous: an Army battalion task force, a Navy carrier battle group, a Marine expeditionary unit, and a joint task force or joint strike package.

Fifth, the DOO serves as the framework by which other entities can be associated. Three types of relationships were introduced: Organization-to-Organization Relationships (OTORs) to define associations between organizations (i.e., for task organizing); Equipment-to-Organization Relationships (ETORs) to define associations between materiel and organizations (called alignments), and Personnel-to-Organization Relationships (PTORs) to define associations between people and organizations—in particular, billets. Therefore, OTORs, ETORs, and PTORs can be used to integrate unit databases with those for personnel and logistics. All this is a result of choosing the concept of task organization as the foundation for these other battle command entities.

Task organization is at the heart of the MCP process. The vision is that each service will maintain an "org server" that contains its DOO in a common form that is readily accessible by the other services. The Army force development proponent, the United States Army Force Management Support Agency (USAFMSA), is currently evaluating this approach via an "Org-ID Pilot Program." By April 2000, four battalions from the 4th Infantry Division, the Army's first Force XXI digitized division, will have been converted into prototype DOOs in an effort to identify major problems or overlooked details in this concept. After that, a decision will be made to continue with the rest of the Army.

5 RECOMMENDATIONS

This report is in itself a recommendation as to how to represent and document military forces to facilitate the digitization process. As a result of this study, the author fully believes that the concept of *task organization* forms the heart of the battle command process. However, if this is true, then there is one readily apparent critical shortcoming: *there is no joint forum by which force developers can assemble to discuss service-specific and joint force developmental issues*. In fact, it appears that there are few processes that differ across the services more than force development. Consequently, it is recommended that the Joint Forces Command assume the initiative to help these disparate service communities become cohesive.

Second, the basic DOO concept appears to be fully extendible to coalition forces. Therefore, it is recommended that the DOO approach be investigated for use in coalition operations. To date, a prototype Org-ID server has been built by the U.S. Army Research Laboratory for the U.S. Army Office of the Director of Information for Command, Control, Communication, and Computers and can be used for evaluation at any time.

Finally, task organization is clearly a joint problem. It is recommended that a common application, perhaps called a "joint task organization toolkit," be developed to allow rapid "Plug and Play" of organizations across service and coalition boundaries. This application could initially operate using the Joint Common Database (JCDB) and could ultimately be a piece of "standard issue" software for all battle command systems.

REFERENCES

Alberts, D.S., "Mission Capability Packages, Strategic Forum, 1-4," Washington, DC: National Defense University Press, January 1995. http://www.dodccrp.org/alberts1.htm

Booch, G., "Object-Oriented Analysis and Design" (second edition). Addison-Wesley Publishing Company, Menlo Park, CA, 1994.

Bruce, T.A., "Designing Quality Databases with IDEF1X Information Models," Dorset House Publishing, New York, NY, 1992.

Chamberlain, S.C., "Technical Foundations for Emerging Battlefield Information Management." *Proceedings of the 4th International Symposium on Command and Control Research and Technology*. Näsby Slott, Sweden, 14 – 16 Sep 1998. http://www.dodccrp.org/Proceedings/DOCS/wcd00000/wcd00086.htm

Chamberlain, S.C., "Default Operational Representations of Military Organizations for Joint and Coalition Operations." *Proceedings of the 1999 Command and Control Research and Technology Symposium.* Naval War College; Newport, RI, 29 Jun – 1 Jul 99. http://www.dodccrp.org/1999CCRTS/pdf_files/track_5/056chamb.pdf

Codd, E.F., "Extending the Database Relational Model to Capture More Meaning." *ACM Transactions on Database Systems*, Vol 4(4), 397-434, 1979.

Date, C.J., "Relational Database: Selected Writing." Addison-Wesley Publishing Company, Reading, MA, 1986.

Grant, J., and J. Minker. "The Impact of Logic Programming on Databases." *Communications of the ACM, Vol 35(3)*, March 1992.

Johnston, T. "Primary Key Reengineering Projects: The Problem" *Data Management Review.* February 2000. Available from: http://www.dmreview.com/master.cfm?NavID=55&EdID=1866.

Knuth, D.E., "The Art of Computer Programming - Vol 1" (second edition). Addison-Wesley Publishing Company, Reading, MA, 1973.

Lonigro, M., "The Case for the Surrogate Key." *Intelligent ENTERPRISE'S DATABASE online* (http://www.dbpd.com), Miller Freeman Inc, 1997.

Minker, J., "Logic and Databases: A 20 Year Retrospective," Invited Keynote Address, *Workshop on Logic in Databases*, San Miniato, Italy, July 1996; Available from: http://karna.cs.umd.edu:3264/papers/Retrospective/Retrospective.html.

Mintzberg, H., and L. Van der Heyden, "Organigraphs: Drawing How Companies Really Work." *Harvard Business Review*, Vol 77(5), September 1999.

Robinson, J., "Logic and Logic Programming." *Communications of the ACM*, Vol 35(3), March 1992.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. "Object-Oriented Modeling and Design." Prentice Hall, Englewood Cliffs, NJ, 1991.

Service Staffing Documents:

U.S. Air Force Unit Manning Documents (UMD):

- 27th Fighter Squadron (27FSQ), Received 4 Sep 1998
- 55th Fighter Squadron (27FSQ), Received 4 Sep 1998
- 552d Air Control Wing (w/ 26 sub-units), Received 16 Sep 1998

U.S. Army Table of Organization and Equipment (TO&E):

- TO&E 07247F000, RFL CO INF BN (MECH) (XXI), Dated 23 Apr 1998
- TO&E 17377F000 TANK COMPANY, TK BN (XXI), Dated 23 Apr 1998
- TO&E 01387F000 ATTACK COMPANY (AH-64), Dated 24 Apr 1998

U.S. Navy Ship/Squadron Manpower Documents (SMD):

- VAQ-141 Squadron Manpower Document, Dated 1 Jul 1998
- DDG 51 Class (51-67) Ship Manpower Document, Dated 8 Jul 1994

U.S. Marine Corps Table of Manpower Requirements (Table of Organization, T/O):

- TO/TE 1101H, HQ Btry, Arty Regt, MarDiv, FMF, Dated 28 Feb 1994
- TO/TE 1142G, HQ Btry, Arty Bn, Arty Regt, MarDiv, FMF, Dated 28 Feb 1994
- TO/TE 1113G, 155mm How Btry, Arty Bn, Arty Regt, MarDiv, FMF, Dated 28 Feb 1994
- Squadron/Company Echelon Summary of I MEF, Dated 13 Sep 1994

Data Models:

DoD C4ISR Core Architecture Data Model (CADM), V 2.0, 1 Dec 1998
http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/index.htm

DoD Command and Control Core Data Model (C2 Core DM),
http://www-datadmn.itsi.disa.mil/ddm.html

ADatP-5 Draft 1.0, *The Land C2 Information Exchange Data Model*, 1 Oct 1999
http://www.euronet.nl/users/atccis/index.html

Joint Common Database (JCDB) – POC: Mr. Robert Carnevale,
Rcarnevale@c3smail.monmouth.army.mil

Appendix A

CURRENT DOD AND NATO DEFINITIONS RELATED TO ORGANIZATIONAL RELATIONSHIPS

INTENTIONALLY LEFT BLANK

CURRENT DOD AND NATO DEFINITIONS RELATED TO ORGANIZATIONAL RELATIONSHIPS

These definitions are from the on-line *DoD Dictionary of Military Terms* located at URL (universal resource locator): http://www.dtic.mil/doctrine/jel/doddict/index.html

1. **Unit** (DoD, NATO)
   1. *Any military element whose structure is prescribed by competent authority, such as a table of organization and equipment; specifically, part of an organization.*
   2. An organization title of a subdivision of a group in a task force.
   3. A standard or basic quantity into which an item of supply is divided, issued, or used. In this meaning, also called unit of issue.
   4. With regard to reserve components of the Armed Forces, denotes a Selected Reserve unit organized, equipped and trained for mobilization to serve on active duty as a unit or to augment or be augmented by another unit. Headquarters and support functions without wartime missions are not considered units.

2. **Organization** – not in the official DoD dictionary (on-line version cited above).

3. **Billet** (DoD)
   1. Shelter for troops.
   2. To quarter troops.
   3. *A personnel <position> or assignment which may be filled by one person.*

4. **Assign** (DoD, NATO)
   1. To place units or personnel in an organization where such placement is relatively permanent, and/or where such organization controls and administers the units or personnel for the primary function, or greater portion of the functions, of the unit or personnel.
   2. To detail individuals to specific duties or functions where such duties or functions are primary and/or relatively permanent. See also *attach*.

5. **Organic** (DoD)
   Assigned to and forming an essential part of a military organization. Organic parts of a unit are those listed in its table of organization for the Army, Air Force, and Marine Corps, and are assigned to the administrative organizations of the operating forces for the Navy.

6. **Order of battle** (DoD, NATO)
   The identification, strength, command structure, and disposition of the personnel, units, and equipment of any military force.

7. **Task-organizing** (DoD)
   The act of designing an operating force, support staff, or logistics package of specific size and composition to meet a unique task or mission. Characteristics to examine when task-organizing the force include, but are not limited to: training, experience, equipage, sustainability, operating environment, enemy threat, and mobility.

8. Attach (DoD)

1. The placement of units or personnel in an organization where such placement is relatively temporary.
2. The detailing of individuals to specific functions where such functions are secondary or relatively temporary, e.g., attached for quarters and rations; attached for flying duty. See also *assign*.

9. Operational authority (DoD)

That authority exercised by a commander in the chain of command, defined further as combatant command (command authority), operational control, tactical control, or a support relationship. See also *combatant command (command authority); in support of; operational control; support; tactical control.*

10. Support (DoD)

1. The action of a force which aids, protects, complements, or sustains another force in accordance with a directive requiring such action.
2. A unit which helps another unit in battle. Aviation, artillery, or naval gunfire may be used as a support for infantry.
3. A part of any unit held back at the beginning of an attack as a reserve.
4. An element of a command which assists, protects, or supplies other forces in combat. See also *close support; direct support; general support; interdepartmental/agency support; international logistic support; inter-Service support; mutual support.*

11. In support of (DoD, NATO)

Assisting or protecting another formation, unit, or organization while remaining under original control.

12. Administrative control (ADCON), (DoD)

Direction or exercise of authority over subordinate or other organizations in respect to administration and support, including organization of Service forces, control of resources and equipment, personnel management, unit logistics, individual and unit training, readiness, mobilization, demobilization, discipline, and other matters not included in the operational missions of the subordinate or other organizations.

13. Operational control (OPCON), (DoD)

Transferable command authority that may be exercised by commanders at any echelon at or below the level of combatant command. Operational control is inherent in combatant command (command authority). Operational control may be delegated and is the authority to perform those functions of command over subordinate forces involving organizing and employing leads and forces, assigning tasks, designating objectives, and giving authoritative direction necessary to accomplish the mission. Operational control includes authoritative direction over all aspects of military operations and joint training necessary to accomplish missions assigned to the command. Operational control should be exercised through the commanders of subordinate organizations. Normally this authority is exercised through subordinate joint force commanders and Service and/or functional component commanders. Operational control normally provides full authority to organize leads and forces and to employ those forces as the commander in operational control considers necessary to accomplish assigned missions. Operational control does not, in and of itself, include authoritative direction for logistics or matters of administration, discipline, internal organization, or unit training. Also called OPCON. See also *combatant command; combatant command (command authority); tactical control.*

14. **Tactical control** (TACON), (DoD)
Command authority over assigned or attached forces or leads, or military capability or forces made available for tasking, that is limited to the detailed and, usually, local direction and control of movements or maneuvers necessary to accomplish missions or tasks assigned. Tactical control is inherent in operational control. Tactical control may be delegated to, and exercised at any level at or below the level of combatant command. Also called TACON. See also *combatant command; combatant command (command authority); operational control.*

15. **Combatant command** (DoD)
A unified or specified command with a broad continuing mission under a single commander established and so designated by the President, through the Secretary of Defense and with the advice and assistance of the Chairman of the Joint Chiefs of Staff. Combatant commands typically have geographic or functional responsibilities. See also *specified command; unified command.*

16. **Combatant command (command authority)** (DoD)
Nontransferable command authority established by title 10 ("Armed Forces"), United States Code, section 164, exercised only by commanders of unified or specified combatant commands unless otherwise directed by the President or the Secretary of Defense. Combatant command (command authority) cannot be delegated and is the authority of a combatant commander to perform those functions of command over assigned forces involving organizing and employing commands and forces, assigning tasks, designating objectives, and giving authoritative direction over all aspects of military operations, joint training, and logistics necessary to accomplish the missions assigned to the command. Combatant command (command authority) should be exercised through the commanders of subordinate organizations. Normally this authority is exercised through subordinate joint force commanders and Service and/or functional component commanders. Combatant command (command authority) provides full authority to organize and employ commands and forces as the combatant commander considers necessary to accomplish assigned missions. Operational control is inherent in combatant command (command authority). Also called *COCOM.* See also *combatant command; combatant commander; operational control; tactical control.*

17. **Primary aircraft authorization** (DoD)
Aircraft authorized to a unit for performance of its operational mission. The primary authorization forms the basis for the allocation of operating resources to include manpower, support equipment, and flying-hour funds. See also backup aircraft authorization.

From NATO Dictionary and Joint Pub 1-02:

18. **Operational command** (OPCOM) (NATO only)
    1. The term is synonymous with operational control and is uniquely applied to the operational control exercised by the commanders of combatant, unified, and specified commands over assigned forces.
    2. The authority granted to a commander to assign missions or tasks to subordinate commanders, to deploy units, to reassign forces, and to retain or delegate operational or tactical control as may be deemed necessary. It does not include responsibility for administration or logistics. OPCOM may also be used to denote the forces assigned to a commander. (See also *operational control (OPCON).*)

19. **Operational control** (OPCON) (JP 1-02) – same as Definition 13.

20. **Tactical control** (TACON) (JP 1-02) [Versus Definition 14.]
    The detailed and, usually, local direction and control of movements or maneuvers necessary to accomplish missions or tasks assigned. (Army) – Tactical control allows commanders below combatant command level to apply force and direct the tactical use of logistics assets but does not provide authority to change organizational structure or direct administrative and logistical support. See FMs 1-111, 31-20, 71-100, 100-15, and 101-5.

A Subset from the NATO Army Tactical Command and Control Information System (ATCCIS) Generic Hub Data Model as documented in: ADatP-5, Draft 1.0, *The Land C2 Information Exchange Data Model*, 1 October 1999.

Domain Name	organisation-organisation-assoc-subcategory-code	
Definition	The specific value that represents or denotes the detailed type of relationship between the subject ORGANISATION and the object ORGANISATION in a specific ORGANISATION-ORGANISATION-ASSOCIATION.	
Definition Source	ATCCIS	
	DOMAIN VALUES	
Value	**Definition**	**Source**
Has organic	The subject ORGANISATION normally has the object ORGANISATION under command in garrison.	ATCCIS
Has assigned	An "object" organisation is placed in the "subject" organisation where such placement is relatively permanent, and/or where such "subject" organisation controls and administers the "object" organisation for its primary functions.	ATP35(B)
Has attached	The placement of units in an organisation where such placement is relatively temporary.	ATP35(B)
Has full command of	The military authority and responsibility of a superior officer to issue orders to subordinates.	AAP-6
Reinforces	The subject ORGANISATION is made available to the object ORGANISATION commander for the purpose of supplementing an in-place force.	ATCCIS
Has in reserve	The object ORGANISATION constitutes a force that may be committed into combat only on the order of the commander of the subject ORGANISATION.	ATCCIS
Has in support of	Term designating the support provided to another unit, formation or organisation while remaining under the initial command.	ATP35(B)
Has as alternate	The "subject" organisation has the "object" organisation as able to execute its functions should the need arise.	ATCCIS
Is the same as	The subject ORGANISATION is deemed to be the same as the object ORGANISATION.	ATCCIS
Plays the role of	The subject ORGANISATION plays the role of an object ORGANISATION.	ATCCIS
Has administrative control of	Direction or exercise of authority over subordinate or other organisations in respect to administrative matters such as personnel management, supply, services, and other matters not included in the operational missions of the subordinate or other organisations.	AAP-6
Has operational command of	The authority granted to a commander to assign missions or tasks to subordinate commanders, to deploy units, to reassign forces, and to retain or delegate operational and/or tactical control as may be deemed necessary.	AAP-6
Has operational control of	The authority delegated to a commander to direct forces assigned so that the commander may accomplish specific missions or tasks which are usually limited by function, time, or location; to deploy units concerned, and to retain or assign tactical control of those units. It does not include authority to assign separate employment of components of the units concerned.	AAP-6
Has tactical command of	The authority delegated to a commander to assign tasks to forces under his command for the accomplishment of the mission assigned by higher authority.	AAP-6
Has tactical control of	The detailed, and, usually, local direction and control of movements or manoeuvres necessary to accomplish missions or tasks assigned.	AAP-6
Has at priority call	A precedence applied to the task of an artillery unit to provide fire to a formation/unit on a guaranteed basis.	ATP35(B)
Has captured	The subject ORGANISATION has taken possession, as a result of forceful means, of the object ORGANISATION.	ATCCIS
Has in direct support	The support provided by a unit not attached to or under command of the supported unit or formation, but required to give priority to the support required by that unit or formation.	AAP-6
Has in general support	The support which is given to the supported force as a whole and not to any particular subdivision thereof.	NATO STANAG 2934
Has reinforcing	In artillery usage, a tactical mission in which one artillery unit augments the fire of another artillery unit.	NATO STANAG 2887
Has in general support reinforcing	General Support Reinforcing artillery has the mission of supporting the forces as a whole and, on a secondary basis, of providing reinforcing fire for another artillery unit.	NATO STANAG 2887

INTENTIONALLY LEFT BLANK

Appendix B

APPROXIMATELY 80% OF NODES ARE LEAVES

INTENTIONALLY LEFT BLANK

APPROXIMATELY 80% OF NODES ARE LEAVES
(or How Big is 2^{32}?)

Approximate the number of organizations in a U.S. Army division.
For simplicity, assume a *symmetrical* tree (all nodes have the same number of children):

How many children should each node of the symmetrical tree have (what is its degree)?

Echelon Name	Tree Level
Division	0
Brigade	1
Battalion	2
Company	3
Platoon	4
Squad	5
Soldier	6

1. Enumerate the levels of the organization tree.
2. Assume each echelon has the same degree (i.e., has the same number of sub-echelons = N).
3. At the bottom level, Level 6 in this example, reside the leaf nodes that correspond to billets (i.e., have a 1:1 correspondence with people). If there are 15,000 to 17,000 people in a division, how big should N be?

   In other words, N^6 = ~15,000 to 17,000 = ~16,000.

$$
\begin{aligned}
N^6 &= \sim 16{,}000 \\
\log(N^6) &= \log(N) \times 6 = \log(\sim 16{,}000) \\
\log(N) &= \log(16{,}000) / 6 = .7 \\
10^{\log(N)} &= 10^{.7} = \mathbf{N} = \mathbf{5.02} = \mathbf{\sim 5}
\end{aligned}
$$

So if N = 5, then 5^6 = 15,625 (billets), which is ~15,000 to 17,000 (people in a division).

So if one has a symmetrical tree structure that represents a unit with seven echelons (six sub-levels), and each level has five sub-echelons, then there are a total of 19,531 nodes in such a structure, of which 15,625 are leaf nodes (billets). In other words, there are about 20,000 organizations in such a division. Note that 80% of the nodes are leaf nodes (15,625 of 19,531), and 20% are internal nodes (3906 of 19,531); this matches the theoretical value for a tree where N = 5.

With N = 5: Echelon		# in Echelon
Division	N^0 =	1
Brigade	N^1 =	5
Battalion	N^2 =	25
Company	N^3 =	125
Platoon	N^4 =	625
Squad	N^5 =	3,125
Soldier	N^6 =	15,625
Total:		19,531

The U.S. Army Signal Center says that there are about 33,000 computers in a digitized division, so this number is about right.

At this rate (i.e., 20,000 organizations per division),
2^{32} or 4.3 billion Org-IDs can support 215,00 Army Divisions!

INTENTIONALLY LEFT BLANK

Appendix C

DEPTH-FIRST SEARCH ALGORITHM TO DERIVE A CHAIN OF COMMAND FROM A COMMAND STRUCTURE

INTENTIONALLY LEFT BLANK

DEPTH-FIRST SEARCH ALGORITHM TO DERIVE A CHAIN OF COMMAND FROM A COMMAND STRUCTURE

Assumptions:

- Command Structure graph is a tree (i.e., is fully connected, with one path between any two nodes).
- Internal nodes may (but do not have to) have a second link, called *is_led_by*, to a leaf node (a billet).
- Billet nodes are identifiable.

Data Structures and Functions

Graph ***CS*** = {V, E} represents a command structure organization chart.
Set ***L*** contains the new links that represent a chain of command organization chart using nodes from V, so Graph ***CC*** = {U, L}, where $U \subset V$ (i.e., set U is a subset of set V).
Algorithm uses two "Last-In, First-Out" (LIFO) stacks named: ***S*** (for command structure), and ***P*** (for current parents).

Let:

The variables ***Root, X, Y,*** and ***CP*** (Current Parent) reference nodes.
Nodes labeled ***M*** are marker nodes that are not part of any graph.
The function ***Push(A, B)*** adds node ***A*** to the top of LIFO stack ***B***.
The function ***Pop(B)*** removes the top node from LIFO stack ***B***.
The function ***Top(B)*** returns the value of the top node on the LIFO stack ***B*** and may be equal to EMPTY.
The function ***Add_link(C, link_type, P, G)*** creates a link between child node C and parent node P of type *link_type* and adds it to the set of links in Graph G.
The function ***Get_leaf_node(K, M)*** finds the node pointed to by link M from node K; if no link, it's = 0.
A node N is ***marked*** if $N \in P$. (i.e., N is an element of P)

Description:

This algorithm conducts a depth-first search of Graph CS, and based upon the *is_led_by* links it finds, produces a new graph CC composed of existing leaf nodes connected by new "reports-to" links.

Algorithm

1) Initialize.
   Root = R ; — Select some node ***R*** from ***CS*** to be the root node and
   Push(R,S) ; — add it to the empty LIFO stack ***S***.

2) **While** LIFO stack ***S*** is not empty, do the following:
   a) ***X = Top(S)*** ; ***Pop(S)*** ;
   b) IF ***X*** is a marked billet, THEN — (marked: $X \in P$)
      Do Nothing (ignore the node)
   c) ELSE IF ***X*** is a unmarked billet OR ***X = M*** (is a Marker Node), THEN
      i) IF ***X = M***, THEN — (is a Marker Node)
         X = Top(P) ; Pop(P) ;
      ii) ***CP = Top(P)*** ;
      iii) IF ***CP*** $\neq 0$; — (stack ***P*** is not empty)
         Add_link(X, reports_to, CP, CC) ; — (Create a new CC graph Link)
   d) ELSE (X must be an internal or non-billet leaf node)
      i) ***Y =Get_leaf_node(X, is_led_by)*** ; — (check for an *is_led_by* link)
      ii) IF ***Y*** $\neq 0$, THEN — (an *is_led_by* link exists)
         Push(Y, P) ; Push(M, S) ;
      iii) Push(children of X, S)

INTENTIONALLY LEFT BLANK

NO. OF COPIES	ORGANIZATION
1	ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	US ARMY RESEARCH LAB ATTN AMSRL CS AS REC MGMT 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	US ARMY RESEARCH LAB ATTN AMSRL CI LL TECH LIB 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	US ARMY RESEARCH LAB ATTN AMSRL DD 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	OSD OUSD (A&T) / ODDRE&E(R) ATTN RJ TREW THE PENTAGON WASHINGTON DC 20301-7100
1	DUSA(OR) ATTN MR HOLLIS 102 ARMY PENTAGON RM 2E660 WASHINGTON DC 20301-0102
3	DIRECTOR RESEARCH OASD(C3I) RM 3E172 THE PENTAGON WASHINGTON DC 20301-6000
1	NAVAL SURFACE WARFARE CTR CODE B07 J PENNELLA 17320 DAHLGREN RD BLDG 1470 RM 1101 DAHLGREN VA 22448-5100
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI THAYER HALL WEST POINT NY 10996-1786
10	HQDA ODISC4 ATTN SAIS PAA S RM 1C634 MR B HABERCAMP 100 ARMY PENTAGON WASHINGTON DC 20310-0107

NO. OF COPIES	ORGANIZATION
10	DIRECTOR TPIO ABCS ATTN ATZL TP 415 SHERMAN AVENUE FT LEAVENWORTH KS 66027-2300
5	TSM MCS/GCCS A ATTN ATZL TSM USA COMBINED ARMS CENTER 415 SHERMAN AVENUE UNIT 5 FT LEAVENWORTH KS 66027
2	TSM ASAS USAIC ATTN ATZS AS FT HUACHUCA AZ 85613-6000
2	TSM CSSCS USACASCOM ATTN ATCL K 2521 A AVE BLDG 11620 FT LEE VA 23801-1701
2	TSM FATDS USAFAS ATTN ATSF FSC 3 FT SILL OK 73503-5600
2	TSM FORCE XXI USAAC ATTN ATZK XXI FT KNOX KY 40121-5000
2	TSM SHORAD USAADAS ATTN ATSA TSM SH FT BLISS TX 79916-3802
2	TSM SOLDIER USAIC&S ATTN ATZB TS FT BENNING GA 31905-5405
2	TSM NM USASC ATTN ATZH NM FT GORDON GA 30905-5000
2	TSM WIN T USASC ATTN ATZH WT FT GORDON GA 30905-5000
2	COMMANDER USAAVNC ATTN ATZQ CDO (STALLSMITH) L AVE BLDG 513 FT RUCKER AL 36362
1	DIRECTOR TPIO SE ATTN ATZL NSC TP 410 KEARNEY AVE FT LEAVENWORTH KS 66027-1306

NO. OF COPIES	ORGANIZATION
1	DIRECTOR TPIO TD ATTN ATSE TPIO USAES FT LEONARD WOOD MO 65473-8929
1	COMMANDER USACAC ATTN ATZL CG 415 SHERMAN AVENUE FT LEAVENWORTH KS 66027-2300
1	HQ TRADOC ATTN ATCD ZA FT MONROE VA 23651
1	HQ TRADOC ATTN ATCD ZC FT MONROE VA 23651
1	HQ TRADOC ATTN ATCD G FT MONROE VA 23651
1	HQ TRADOC ATTN ATCD F FT MONROE VA 23651
5	USAFMSA RDD ATTN MOFI FMR PT 415 SHERMAN AVE UNIT 7 FT LEAVENWORTH KS 66027-2300
5	USAFMSA RDD LEE ATTN MOFI FMR L 700 QUARTERS ROAD SUITE 349 FT LEE VA 23801-1703
5	DIRECTOR FDD ATTN ATCD F 415 SHERMAN AVE FT LEAVENWORTH KS 66027-2300
5	USAFMSA ADD ATTN MOFI FMA 9900 BELVOIR ROAD SUITE 120 FT BELVOIR VA 22060-5578
5	USAFMSA IMD ATTN MOFI ZC IM 401 ARMY PENTAGON WASHINGTON DC 20310-0401
5	USAFMSA HQDA SPPT DIV ATTN MOFI ZC SAM 401 ARMY PENTAGON WASHINGTON DC 20310-0401

NO. OF COPIES	ORGANIZATION
1	ARMY FORCE MGMT SCHOOL 5500 21ST STREET SUITE 1400 FT BELVOIR VA 22060-5923
1	DIRECTOR ARMY FORCE PROGRAMS ATTN DAMO FDZ ROOM 3A522 460 ARMY PENTAGON WASHINGTON DC 20310-0460
2	DIRECTOR C2D ATTN AMSEL RD C2 D FORT MONMOUTH NJ 07703
2	US ARMY CERDEC ATTN AMSEL RD C2 O SS (NIEMELA) FORT MONMOUTH NJ 07703
2	US ARMY CERDEC ATTN AMSRL ST DD (SALIS) FORT MONMOUTH NJ 07703
2	PEO C3S ATTN SFAE C3S (BG BOUTELLE) MYER CENTER FT MONMOUTH NJ 07703
5	PEO C3S ATTN SFAE C3S HTI (CARNEVALE) MYER CENTER FT MONMOUTH NJ 07703
2	PM ATCCS ATTN SFAE C3S AT MYER CENTER FT MONMOUTH NJ 07703
5	PM ATCCS CS ATTN SFAE C3S AT CS (USECHAK) MYER CENTER FT MONMOUTH NJ 07703
2	PM MCS ATTN SFAE C3S AT MCS FT MONMOUTH NJ 07703
2	PEO C3S TOC/AMDCCS ATTN SFAE C3S AD REDSTONE ARSENAL AL 35898
2	PM STCCS ATTN SFAE C3S STR 6052 MEADE RD SUITE 101 FORT BELVOIR VA 22060-5260

NO. OF COPIES	ORGANIZATION
2	PM INTEL FUSION ATTN SFAE C3S INT 1616 ANDERSON ROAD MCLEAN VA 22102-1616
2	PM FBCB2 ATTN SFAE C3S AP BLDG 2525 FT MONMOUTH NJ 07703
1	PM FATDS ATTN SFAE C3S FS BLDG 457 FT MONMOUTH NJ 07703-5404
2	PM FATDS ATTN SFAE C3S FS TMD (SAPHOW) BLDG 457 FT MONMOUTH NJ 07703-5404
2	PM WIN T ATTN SFAE C3S WIN FORT MONMOUTH NJ 07703
2	DIRECTOR ADO ATTN DAMO ADO (MR LEVINE) 400 ARMY PENTAGON WASHINGTON DC 20310-0400
1	COMMANDER USAREUR&7 ARMY ATTN AEACG (GEN MEIGS) APO AE 09014
1	HQ USJFCOM ATTN DCINC (LTG BURNETTE) 1562 MITSCHER AVE NORFOLK VA 23551
5	COMMANDER USAF AC2ISRC ATTN CCT (MR BARRINGER) 130 ANDREWS ST SUITE 216 LANGLEY AFB VA 23665
5	NAVY WARFARE DEV CMD CODE N341 ATTN LCDR ESPE 686 CUSHING ROAD NEWPORT RI 02841
2	NAVAL POSTGRADUATE SCHOOL CODE SM/HH (CDR HATCH) 555 DYER ROAD RM 236 MONTEREY CA 93940-5103

NO. OF COPIES	ORGANIZATION
2	SPACE AND NAVAL WARFARE SYSTEMS CENTER ATTN LES ANDERSON D4123 53560 HULL ST SAN DIEGO CA 92152-5001
2	MR CARL A DEFRANCO JR PE AFRL/IFSA 525 BROOKS ROAD SUITE 1015 ROME NY 13441-4505
2	MR JAMES SALTON ATTN SFAE C3S FIO FT MONMOUTH NJ 07703
2	USA LOG INTEGRATION AGENCY ATTN LOIA CD (M BLACKMAN) 5001 EISENHOWER AVENUE ALEXANDRIA VIRGINIA 22333-0001
3	US ARMY RESEARCH LAB ATTN AMSRL IS J GANTT AMSRL IS C J GOWENS AMSRL IS R SLIFE 2800 POWDER MILL RD ADELPHI MD 20783-1197
4	INST FOR DEFENSE ANALYSIS ATTN DR F LOAIZA 1801 N BEAUREGARD ST ALEXANDRIA VA 22311
4	MR H LAVENDER AMERIND INC 1310 BRADDOCK PLACE ALEXANDRIA VA 22333
1	COL DAVE MEASELS (RET) SAIC, MAIL STOP 3 8 1 1710 GOODRIDGE DRIVE MCLEAN VA 22102
2	TOM CAHILL COMPUTER SCIENCES CORP 1301 VIRGINIA DRIVE 1ST FLOOR FORT WASHINGTON PA 19034
2	COL J BESSLER (RET) SPRINT MAILSTOP KSOPAR0301 10895 GRANDVIEW OVERLAND PART KS 66210

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
2	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CI LP (TECH LIB) BLDG 305 APG AA
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CI N RADHAKRISHNAN BLDG 394
11	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL IS C B COOPER S CHAMBERLAIN (10 CPS) BLDG 1116A
10	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL IS CI B BROOME G HARTWIG F BRUNDICK H CATON J BRAND R HELFMAN J DUMER T HANRATTY E BAUER A BRODEEN BLDG 1116A
	ABSTRACT ONLY
1	US ARMY RESEARCH LAB ATTN AMSRL CS EA TP TECH PUB 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DEPUTY CG FOR RDE HQ US ARMY MATERIEL CMD ATTN AMCRD 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001

NO. OF COPIES	ORGANIZATION

MEMBERS OF TTCP C3I GROUP TP-10

2	MR JOHN LAWS (CHAIRMAN) CIS SYSTEMS DEPT DEFENCE EVALUATION & RESEARCH AGENCY ST ANDREWS ROAD MALVERN WORCS WR14 3PS
2	MR JOHN TINDLE (UK NATL LDR) CIS SYSTEMS DEPT DEFENCE EVALUATION & RESEARCH AGENCY ST ANDREWS ROAD MALVERN WORCS WR14 3PS
2	DR JOHN ROBINSON (CA NATL LDR) COMMUNICATIONS RESEARCH CENTRE (CRC) CANADA 3701 CARLING AVENUE BOX 11490 STATION H OTTAWA K2H 8S2 CANADA
2	DR ALLAN GIBB (CA ARMY MBR) INFORMATION SYSTEMS TECHNOLOGY SECTION DREV 2459 PIE XI NORTH BLVD (CP 8800) VAL BELAIR X4683 QUEBEC G3J 1X5 CANADA
2	DR IAIN MACLEOD (AU NATL LDR) DSTO C3 RESEARCH CENTER FERN HILL PARK DEPT OF DEFENCE CANBERRA ACT 2600 AUSTRALIA

INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 2000	Final

4. TITLE AND SUBTITLE

Default Operational Representations of Military Organizations

5. FUNDING NUMBERS

PR: 9T1110

6. AUTHOR(S)

Chamberlain, S.C. (ARL)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Information Science & Technology Directorate
ATTN: AMSRL-IS-C
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DoD Cooperative C4ISR Research Program (CCRP)
Director, Research, OASD(C3I)
Rm 3E172, The Pentagon
Washington, DC 20301-6000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARL-TR-2172

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report is a result of a study commissioned under the auspices of the C4ISR[a] Cooperative Research Program (CCRP) within the OASD(C3I)[b] to attempt to identify why information integration between battlefield systems has not progressed at the rate expected. It was observed that sometimes the most fundamental issues of battle command are overlooked, especially when automation is involved. This is because of one's natural instincts to automate manual procedures rather than exploit the true capabilities of new technologies. This report addresses one of these cases: the identification of many entities inside many interconnected computers, sometimes called the "naming problem." It is argued that this problem strikes at the heart of battle command automation process and, consequently, the development and execution of mission capability packages. However, one must not be deluded into believing that this is merely a computer science problem; it is primarily a military science problem with some computer science technology "sprinkled in." The thesis is presented that the concept of organization (or task organization) is the central theme around which all battle command representations revolve. In essence, the organizational structure forms a framework to which all other battlefield entities can be related, making the organizational data structures the rallying point for the integration of other databases, such as logistics, personnel, and communications. Further, it is shown that fluid orders of battle (OOBs) can most always be built by re-linking existing organizations that are part of a stable default organizational structure. For this to be effective, however, the default structure must include more nodes than provided by current staffing documents. The concept of default operational organizations (DOOs) is introduced to provide a representation of military organizations that meets the requirements necessary to build arbitrary OOBs across joint service and international boundaries. By looking closely at how each service organizes for combat, the author developed basic tenets in an attempt to reduce many practices to a few fundamental concepts. The result is a set of guidelines based upon the "best practices" of all the services, which is used to build DOOs that closely reflect the way we fight or deploy. An architecture is described to maintain the DOO in organization servers, and a unique identifier is proposed, called an organizational identifier (Org-ID), to facilitate the naming process. Org-IDs are distributed by a set of Org-ID servers that guarantee the uniqueness of these identifiers across service and coalition boundaries. Finally, a naming convention is introduced that builds upon the uniqueness of Org-IDs to provide universal surrogate keys for any battlefield entity.

[a] command, control, communications, computers, intelligence, surveillance, and reconnaissance
[b] Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence

14. SUBJECT TERMS

information management	organization	surrogate key
key management	Org-ID	

15. NUMBER OF PAGES

95

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102